

Capítulo 6

Autómatas a pila embebidos

En este capítulo se presentan los autómatas a pila embebidos, el primer modelo de autómatas descrito en la literatura que reconocía exactamente la clase de los lenguajes de adjunción de árboles. Las principales aportaciones de este capítulo son la redefinición de este tipo de autómatas, eliminando la necesidad de un control finito y modificando en consecuencia la forma de las transiciones, y el establecimiento de una técnica de tabulación que permite su ejecución en tiempo polinomial. Este capítulo está basado en [18].

6.1. Introducción

Un autómata para un formalismo gramatical puede simular una derivación en dicho formalismo básicamente de dos maneras: descendente o ascendente. En el caso descendente, las transiciones del autómata tratan de recorrer el árbol derivado desde la raíz hacia las hojas, mientras comprueba si la cadena de entrada concuerda con la cadena esperada. En el caso ascendente, la cadena de entrada se va introduciendo en el almacenamiento del autómata al tiempo que se trata de recorrer el árbol derivado desde las hojas hacia la raíz. En el caso de los autómatas a pila, que constituyen el modelo de autómata equivalente a las gramáticas independientes del contexto, el comportamiento descendente o ascendente viene especificado por el conjunto de transiciones utilizados, no por la forma en que los autómatas a pila son definidos en tanto que sistema formal. No ocurre lo mismo en el caso de los autómatas a pila embebidos, que constituyen uno de los modelos de autómata equivalentes a las gramáticas de adjunción de árboles. Los autómatas a pila embebidos sólo pueden simular derivaciones de una gramática de adjunción de árboles en las que las adjunciones se reconocen de modo descendente, y no pueden ser utilizados para simular derivaciones en las cuales las derivaciones se reconocen de modo ascendente.

6.2. Autómatas a pila embebidos

Los autómatas a pila embebidos (*Embedded Push-Down Automata*, EPDA) [218, 206] son una extensión de los autómatas a pila que reconocen exactamente la clase de los lenguajes de adjunción de árboles. La principal diferencia entre un PDA y un EPDA estriba en que mientras los primeros trabajan con una pila de símbolos elementales, los segundos trabajan con una pila, que denominamos pila principal, que a su vez contiene pilas.

Un autómata a pila embebido consta de tres componentes: una cadena de entrada, un control finito y una pila que contiene pilas. Su *configuración* en un momento dado viene determinada por el estado del control finito en que se encuentra el autómata, el contenido de la pila y la parte de la cadena de entrada que resta por leer, tal y como se muestra en la figura 6.1. Un conjunto

de *transiciones* permite cambiar de configuración. Para ello, las transiciones pueden consultar el estado q del control finito, el siguiente símbolo terminal a de la cadena de entrada y el elemento Z en la cima de la pila $[\alpha Z$ que se encuentra en la cima de la pila principal. Como resultado de la aplicación de una transición, el autómata puede cambiar al estado q' , avanzar una posición en la cadena de entrada y reemplazar Z por una secuencia $Z_m \dots Z_1$ de símbolos de pila para dar lugar a una pila $[\alpha'$ en la cima de la pila principal. Adicionalmente, la pila $[\alpha'$ puede ser reemplazada por una secuencia de $k \geq 0$ pilas, que incluirán a la pila $[\alpha'$ si esta no es la vacía. La figura 6.2 muestra el resultado de aplicar una transición

$$(q', [\alpha_k \dots [\alpha_{i+1}, Z_m \dots Z_1, [\alpha_i \dots [\alpha_1) \in \delta(q, a, Z)$$

al autómata a pila embebido de la figura 6.1. Esta transición reemplaza el elemento Z de la cima de la pila $[\alpha Z$ por $Z_m \dots Z_1$ para dar lugar a la pila $[\alpha Z_m \dots Z_1$. Bajo dicha pila se sitúan $k - i$ pilas $[\alpha_j$ con $i + 1 \leq j \leq k$. Por encima se sitúan i pilas $[\alpha_j$ con $1 \leq j \leq i$. La cima de $[\alpha_1$ se convierte en la cima de la pila principal. El marcador $[$ no pertenece a V_S y se utiliza simplemente para separar el contenido de las diferentes pilas.

Si como resultado de la aplicación de una transición la pila situada en la cima de la pila principal quedase vacía, dicha pila sería eliminada y la pila situada inmediatamente debajo pasaría a ocupar la cima de la pila principal.

De acuerdo con Vijay-Shanker en [206], definiremos formalmente un autómata a pila embebido como una tupla $(Q, V_T, V_S, \delta, q_0, Q_F, \$_0)$ donde:

- Q es un conjunto finito de estados.
- V_T es un conjunto finito de símbolos terminales.
- V_S es un conjunto finito de símbolos de pila.
- q_0 es el estado inicial.
- $Q_F \subseteq Q$ es el conjunto de estados finales.
- $\$_0 \in V_S$ es el símbolo inicial de la pila.
- δ es una relación de $Q \times V_T \cup \{\epsilon\} \times V_S$ en subconjuntos finitos de $Q \times ([V_S^*]^* \times V_S^* \times ([V_S^*]^* \times V_S^* \times V_S^*))$, donde $[\notin V_S$ es un símbolo utilizado para separar las diferentes pilas que componen la pila principal.

La *configuración* de un autómata a pila embebido en un momento dado viene definida por el triple (q, Υ, w) , donde $q \in Q$ indica el estado en el que se encuentra, $\Upsilon \in ([V_S^*]^*)^*$ el contenido de la pila principal y $w \in V_T^*$ la parte de la cadena de entrada que resta por leer. El cambio de una configuración a otra viene determinado por la aplicación de una transición, de tal modo que si $(q, \Upsilon[\alpha Z, aw)$ es una configuración y

$$(q', [\alpha_k \dots [\alpha_{i+1}, Z_m \dots Z_1, [\alpha_i \dots [\alpha_1) \in \delta(q, a, Z)$$

es una transición, entonces el autómata a pila embebido pasa a la nueva configuración

$$(q', \Upsilon[\alpha_k \dots [\alpha_{i+1}[\alpha Z_m \dots Z_1[\alpha_i \dots [\alpha_1, w)$$

Este hecho se denota mediante

$$(q, \Upsilon[\alpha Z, aw) \vdash (q', \Upsilon[\alpha_k \dots [\alpha_{i+1}[\alpha Z_m \dots Z_1[\alpha_i \dots [\alpha_1, w)$$

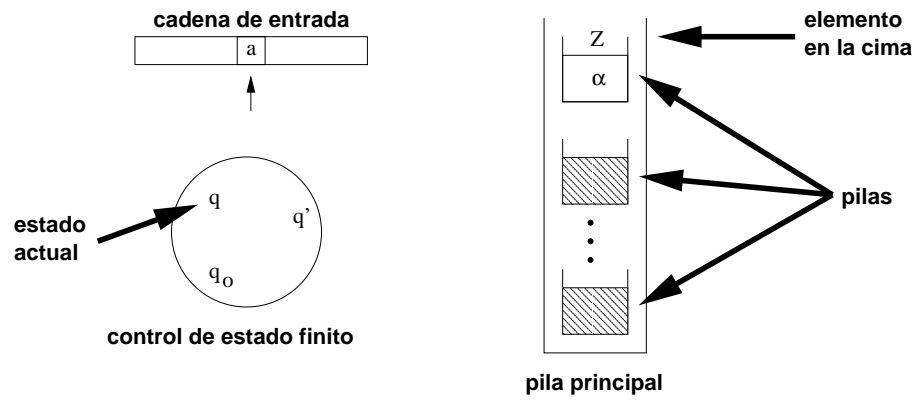


Figura 6.1: Autómata a pila embebido

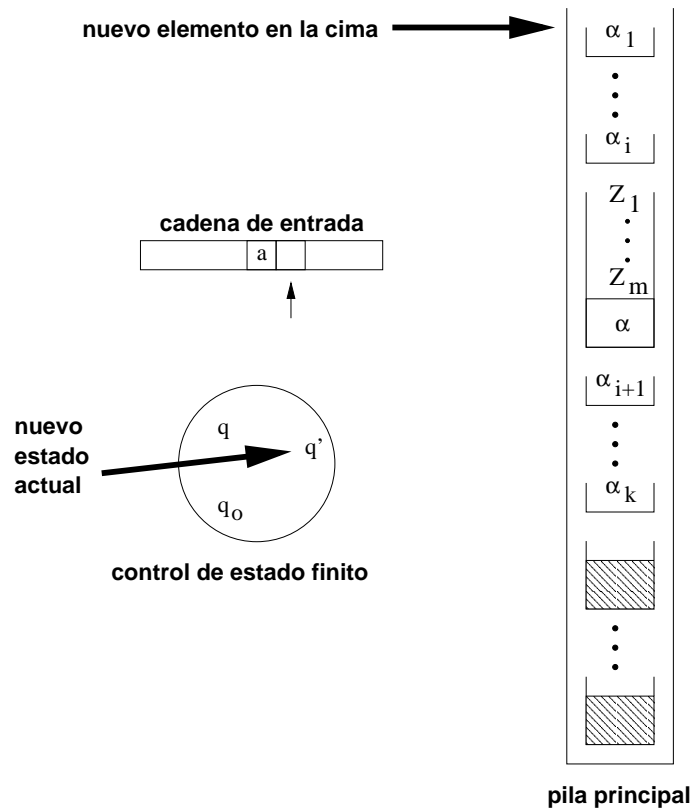


Figura 6.2: Autómata a pila embebido después de una transición

Denotamos por \vdash^* el cierre reflexivo y transitivo de \vdash .

El lenguaje aceptado por estado final por un autómata a pila embebido viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $(q_0, [\$_0, w]) \vdash^* (p, \Upsilon, \epsilon)$, donde $p \in Q_F$ y $\Upsilon \in ([V_S^*])^*$.

El lenguaje aceptado por pila vacía por un autómata a pila embebido viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $(q_0, [\$_0, w]) \vdash^* (q, \epsilon, \epsilon)$ para cualquier $q \in Q$.

Dado un EPDA que reconoce un determinado lenguaje por estado final, es posible construir otro EPDA que reconoce el mismo lenguaje por pila vacía y viceversa [206].

Ejemplo 6.1 El autómata embebido a pila definido por la tupla $(\{q_0, q_1, q_2, q_3\}, \{a, b, c, d\}, \{\$_0, B, C, D\}, \delta, q_0, \emptyset, \$_0)$, donde δ contiene las transiciones mostradas en la tabla 6.1, acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$ por pila vacía. En la tabla 6.1 también se muestra la secuencia de configuraciones que sigue el autómata para analizar correctamente la cadena de entrada $aabbccdd$. La primera columna muestra la transición aplicada, la segunda el estado, la tercera el contenido de la pila y la cuarta la parte que resta por leer de la cadena de entrada. ¶

(a)	$(q_0, [D, B, \epsilon] \in \delta(q_0, a, \$_0)$	q_0	$[\$_0$	$aabbccdd$	
(b)	$(q_0, [D, BB, \epsilon] \in \delta(q_0, a, B)$	(a)	q_0	$[D[B$	$abbccdd$
(c)	$(q_1, [C, \epsilon, \epsilon] \in \delta(q_0, b, B)$	(b)	q_0	$[D[D[BB$	$bbccdd$
(d)	$(q_1, [C, \epsilon, \epsilon] \in \delta(q_1, b, B)$	(c)	q_1	$[D[D[C[B$	$bccdd$
(e)	$(q_2, \epsilon, \epsilon, \epsilon] \in \delta(q_1, c, C)$	(d)	q_1	$[D[D[C[C$	$ccdd$
(f)	$(q_2, \epsilon, \epsilon, \epsilon] \in \delta(q_2, c, C)$	(e)	q_2	$[D[D[C$	cdd
(g)	$(q_3, \epsilon, \epsilon, \epsilon] \in \delta(q_2, d, D)$	(f)	q_2	$[D[D$	dd
(h)	$(q_3, \epsilon, \epsilon, \epsilon] \in \delta(q_3, d, D)$	(g)	q_3	$[D$	d
(i)	$(q_0, \epsilon, \epsilon, \epsilon] \in \delta(q_0, \epsilon, \$_0)$	(h)	q_3		

Tabla 6.1: Transiciones del autómata a pila embebido que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ (izquierda) y configuraciones de dicho autómata durante el análisis de $aabbccdd$ (derecha)

Johnson define en [89] una extensión de los autómatas a pila embebidos en la cual los símbolos de pila son reemplazados por términos lógicos de primer orden. Johnson denomina a tal extensión *autómatas lógicos a pila embebidos* (*Logic Embedded Push-Down Automata*, LEPDA) y propone una implementación en el lenguaje de programación lógica Prolog [225].

Weir generaliza en [231] el concepto de autómata a pila embebido al definir una progresión de modelos de autómatas, denominada *autómatas a pila iterados lineales* en la cual los autómatas

de nivel 0, que no utilizan pila, aceptan la clase de los lenguajes regulares; los autómatas de nivel 1, que utilizan una pila, aceptan la clase de los lenguajes independientes del contexto; y los autómatas de nivel 2, que hacen uso de una pila de pilas, aceptan la clase de los lenguajes de adjunción de árboles.

6.3. Autómatas a pila embebidos sin estados

Al igual que en el caso de los autómatas a pila, el control finito es un elemento prescindible de los autómatas a pila embebidos, puesto que el estado correspondiente a una configuración puede ser incluido en el elemento de la cima de la pila. Como resultado obtenemos una definición alternativa, que juzgamos más simple y homogénea, según la cual un autómata a pila embebido es una tupla $(V_T, V_S, \Theta, \$_0, \$_f)$ en la cual:

- V_T es un conjunto finito de símbolos terminales.
- V_S es un conjunto finito de símbolos de pila.
- $\$_0 \in V_S$ es el símbolo inicial de pila.
- $\$_f \in V_S$ es el símbolo final de pila.
- Θ es un conjunto de transiciones, cada una de las cuales pertenece a uno de los siguientes tipos, donde $C, F, G \in V_S$, $\Upsilon \in ([V_S^*])^*$, $\alpha \in V_S^*$ y $a \in V_T \cup \{\epsilon\}$:

SWAP: Transiciones de la forma $C \xrightarrow{a} F$ que reemplazan el elemento C de la cima de la pila por el elemento F mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C$ es una pila $\Upsilon[\alpha F$.

PUSH: Transiciones de la forma $C \xrightarrow{a} CF$ que apilan un nuevo elemento F en la pila mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C$ es una pila $\Upsilon[\alpha CF$.

POP: Transiciones de la forma $CF \xrightarrow{a} G$ que eliminan los dos elementos C y F de la cima de la pila y los sustituyen por G mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha CF$ es una pila $\Upsilon[\alpha G$.

WRAP-A: Transiciones de la forma $C \xrightarrow{a} C, [F$ que sitúan una nueva pila $[F$ en la cima de la pila principal mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C$ es una pila $\Upsilon[\alpha C[F$.

WRAP-B: Transiciones de la forma $C \xrightarrow{a} [C, F$ que sitúan una nueva pila $[C$ bajo la pila que actualmente ocupa la cima de la pila principal y cambian C por F en la cima de dicha pila, todo ello mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C$ es una pila $\Upsilon[C[\alpha F$.

UNWRAP: Transiciones de la forma $C, [F \xrightarrow{a} G$ que eliminan la pila $[F$ de la cima de la pila principal, mientras se lee a de la cadena de entrada. El resultado de aplicar una transición de este tipo a una pila $\Upsilon[\alpha C[F$ es una pila $\Upsilon[\alpha G$.

Los tres primeros tipos se corresponden con los tipos de transiciones presentes en los autómatas a pila, aunque en este caso trabajan en la pila situada en la cima de la pila principal, tal y como se observa en la figura 6.3. Las transiciones de tipo WRAP-A y WRAP-B permiten situar una nueva pila encima y debajo, respectivamente, de la pila situada en la cima de la pila principal, como se puede observar en la figura 6.4. Las transiciones de tipo UNWRAP permiten eliminar una pila de la cima de la pila principal, como muestra la figura 6.5.

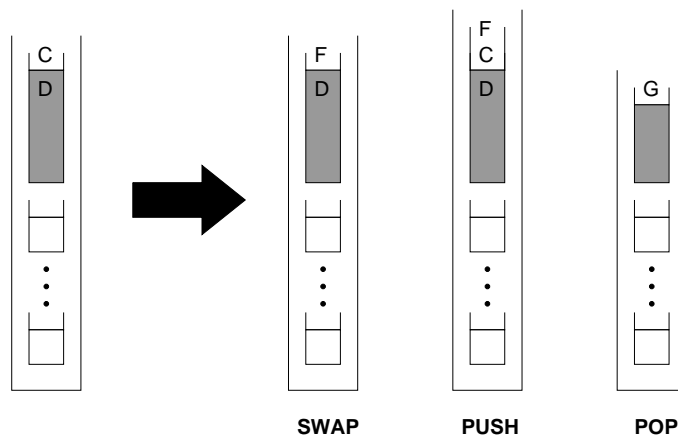


Figura 6.3: Transiciones SWAP, PUSH y POP

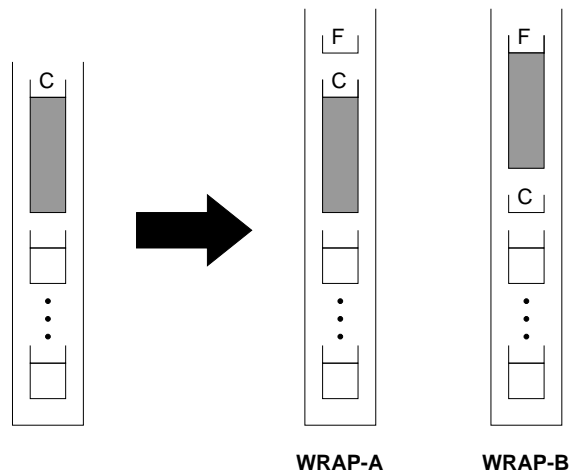


Figura 6.4: Transiciones WRAP-A y WRAP-B

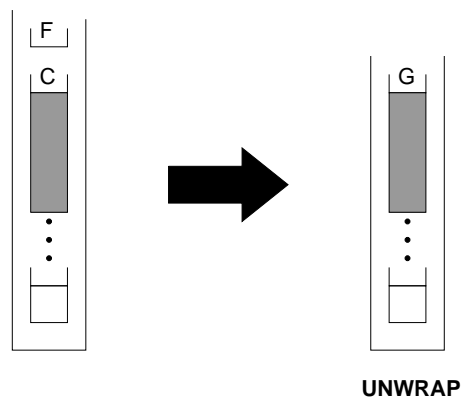


Figura 6.5: Transición UNWRAP

La *configuración* en un momento dado de un autómata a pila embebido sin estados viene determinada por el par (Υ, w) , donde Υ es el contenido de la pila y w la parte de la cadena de entrada que resta por leer. Una configuración (Υ, aw) deriva una configuración (Υ', w) , hecho que denotamos mediante $(\Upsilon, aw) \vdash (\Upsilon', w)$, si y sólo si existe una transición que aplicada a Υ devuelve Υ' y consume a de la cadena de entrada. En caso de ser necesario identificar una derivación d concreta, utilizaremos la notación \vdash_d . Denotamos por \vdash^* el cierre reflexivo y transitivo de \vdash .

Decimos que una cadena de entrada w es aceptada por un autómata a pila embebido si $([\$_0, w) \vdash^* ([\$_0[\$_f, \epsilon)$. El *lenguaje aceptado* por un autómata a pila embebido viene determinado por el conjunto de cadenas $w \in V_T^*$ tal que $([\$_0, w) \vdash^* ([\$_0[\$_f, \epsilon)$.

En lo que resta, cuando nos refiramos a EPDA sin especificar si se trata de la versión con estados o sin estados, debe entenderse que nos estamos refiriendo a la versión sin estados que acabamos de definir.

Ejemplo 6.2 El autómata embebido a pila sin estados definido por la tupla $(\{a, b, c, d\}, \{\$0, \$f, B, C, D, E\}, \Theta, \$0, \$f)$, donde Θ contiene las transiciones mostradas en la tabla 6.2, acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$ por pila vacía. En la tabla 6.2 también se muestra la secuencia de configuraciones que sigue el autómata para analizar correctamente la cadena de entrada $aabbccdd$. La primera columna muestra la transición aplicada, la segunda el contenido de la pila y la tercera la parte que resta por leer de la cadena de entrada. ¶

Teorema 6.1 *Mediante la utilización de un conjunto de transiciones SWAP, PUSH, POP, WRAP-A, WRAP-B y UNWRAP es posible emular transiciones complejas del tipo*

$$DB \xrightarrow{a} [F_k \dots [F_{i+1}, EC_1 \dots C_m, [F_i \dots [F_1$$

tal que su aplicación da lugar a un paso de derivación

$$\Upsilon[\alpha DB \vdash \Upsilon[F_k \dots [F_{i+1}[\alpha EC_1 \dots C_m, [F_i \dots [F_1$$

donde

- $a \in V_T \cup \{\epsilon\}$.
- $0 \leq m \leq 2$.
- $B, C_1, \dots, C_2, F_1, \dots, F_k \in V_S$.
- Si $m = 2$ entonces $C_1 = B$.
- Si $m = 0$ entonces $D, E \in V_S$.
- Si $m > 0$ entonces $D = E = \epsilon$.

Demostración:

Realizaremos una prueba constructiva creando un procedimiento que permita emular el tipo de transiciones propuesto en el teorema 6.1 a partir de las transiciones elementales de los autómatas a pila embebidos sin estados. Para ello precisamos nuevos símbolos de pila ∇_j y X' , donde $i + 1 \leq j \leq k$ y $X \in V_S$. La emulación se realizará en tres fases: la primera se encargará de crear las pilas $[F_k \dots [F_{i+1}$, la segunda de obtener la pila $[\alpha EC_1 \dots C_m$ y la tercera se encargará de crear las pilas $[F_i \dots [F_1$.

(a) $\$0 \mapsto \$0, [D$	$[\$0$	$aabbccdd$
(b) $D \xrightarrow{a} [D, B$	(a) $[\$0[D$	$aabbccdd$
(c) $B \mapsto BD$	(b) $[\$0[D[B$	$abbccdd$
(d) $B \mapsto BC$	(c) $[\$0[D[BD$	$abbccdd$
(e) $C \xrightarrow{b} [C, E$	(b) $[\$0[D[D[BB$	$bbccdd$
(f) $BE \mapsto C$	(d) $[\$0[D[D[BBC$	$bbccdd$
(g) $C, [C \xrightarrow{c} C$	(e) $[\$0[D[D[C[BBE$	$bccdd$
(h) $D, [C \xrightarrow{d} D$	(f) $[\$0[D[D[C[BC$	$bccdd$
(i) $D, [D \xrightarrow{d} D$	(e) $[\$0[D[D[C[C[BE$	$ccdd$
(j) $D \mapsto \$f$	(f) $[\$0[D[D[C[C[C$	$ccdd$
	(g) $[\$0[D[D[C[C$	cdd
	(g) $[\$0[D[D[C$	dd
	(h) $[\$0[D[D$	d
	(i) $[\$0[D$	
	(j) $[\$0[\f	

Tabla 6.2: Transiciones del EPDA sin estados que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ (izquierda) y configuraciones de dicho autómata durante el análisis de $aabbccdd$ (derecha)

Fase 1 Partimos de una configuración $(\Upsilon[\alpha DB, aw)$, donde $D \in V_S$ si $m = 0$ y $D = \epsilon$ en otro caso. Comenzamos la emulación mediante la creación de una transición que apila ∇_k :

$$B \mapsto B\nabla_k$$

Para emular la creación de cada una de las $k - i$ pilas unitarias creamos $k - i$ conjuntos con las tres transiciones siguientes:

$$\begin{aligned} \nabla_j &\mapsto \nabla_j F_j \\ F_j &\mapsto [F_j, F'_j \\ \nabla_j F'_j &\mapsto \nabla_{j-1} \end{aligned}$$

donde $i + 1 \leq j \leq k$.

En el caso de que $k = i$, esto es $[F_k \dots [F_{i+1} = \epsilon$, esta fase consistiría únicamente de la transición

$$B \mapsto B\nabla_i$$

Como resultado de la aplicación de esta fase obtendremos una configuración

$$(\Upsilon[F_k \dots [F_{i+1}[\alpha DB\nabla_i, aw)$$

Fase 2 Las transiciones de esta fase dependen del valor de m , por lo que tenemos tres posibilidades:

- Si $m = 2$ se trata del apilamiento de C_2 sobre B :

$$\nabla_i \mapsto C'_2$$

- Si $m = 1$ nos encontramos ante el cambio de B por C_1 :

$$B\nabla_i \mapsto C'_1$$

- Si $m = 0$ se trata de eliminar B de la pila:

$$B\nabla_i \mapsto \nabla''_i$$

$$D\nabla''_i \mapsto E'$$

Después de esta fase obtenemos una configuración $(\Upsilon[F_k \dots [F_{i+1}[\alpha'X', aw)$, donde $\alpha'X' = \alpha E'$ si $m = 0$, $\alpha'X' = \alpha C'_1$ si $m = 1$ y $\alpha'X' = \alpha BC'_2$ si $m = 2$.

Fase 3 Para iniciar esta fase precisamos de una transición

$$X' \mapsto X', [F'_i$$

donde $X' = E'$ si $m = 0$, $X' = C'_1$ si $m = 1$ y $X' = C'_2$ si $m = 2$. Para crear las restantes $i - 1$ pilas unitarias utilizaremos $i - 1$ transiciones de la forma

$$F'_j \mapsto F'_j, [F'_{j-1}$$

donde $i \leq j < 1$, para finalizar con la transición

$$F'_1 \xrightarrow{a} F_1$$

En el caso de que $i = 0$, esto es $F_i \dots [F_1 = \epsilon$, esta fase consistiría únicamente de la siguiente transición

$$X' \xrightarrow{a} X$$

donde $X = E$ si $m = 0$, $X = C_1$ si $m = 1$ y $X = C_2$ si $m = 2$.

Tras esta fase obtendremos la configuración

$$(\Upsilon[F_k \dots [F_{i+1}[\alpha X' [F'_i \dots [F_1, w)$$

Para emular completamente la transición deseada, deberemos crear una transición

$$F'_j, [W \mapsto Y$$

por cada transición $F'_j, [W \mapsto Y$ presente en el EPDA y una transición

$$X', [W \mapsto Y$$

por cada transición $X, [W \mapsto Y$ presente en el EPDA, donde $X = E$ si $m = 0$, $X = C_1$ si $m = 1$ y $X = C_2$ si $m = 2$.

□

Ejemplo 6.3 La transición $B \xrightarrow{a} [E[F, BC, [G[H$ se emula mediante el conjunto de transiciones mostrado en la tabla 6.3 más una transición $C'[X \mapsto Y$ por cada transición $C[X \mapsto Y$ presente en el EPDA original. Análogamente, deberemos crear una transición $G'[X \mapsto Y$ por cada transición $G[X \mapsto Y$ presente en el EPDA original.

En la misma tabla se muestra el resultado de aplicar las transiciones resultantes a una configuración $(\Upsilon[\alpha B, aw)$. La primera columna muestra la transición aplicada, la segunda muestra el contenido de la pila del EPDA y la tercera la parte de la cadena de entrada que falta por leer.

¶

(a) $B \mapsto B\nabla_4$	$\Upsilon[\alpha B$	aw
(b) $\nabla_4 \mapsto \nabla_4 E$	(a) $\Upsilon[\alpha B\nabla_4$	aw
(c) $E \mapsto [E, E'$	(b) $\Upsilon[\alpha B\nabla_4 E$	aw
(d) $\nabla_4 E' \mapsto \nabla_3$	(c) $\Upsilon[E[\alpha B\nabla_4 E'$	aw
(e) $\nabla_3 \mapsto \nabla_3 F$	(d) $\Upsilon[E[\alpha B\nabla_3$	aw
(f) $F \mapsto [F, F'$	(e) $\Upsilon[E[\alpha B\nabla_3 F$	aw
(g) $\nabla_3 F' \mapsto \nabla_2$	(f) $\Upsilon[E[F[\alpha B\nabla_3 F'$	aw
(h) $\nabla_2 \mapsto C'$	(g) $\Upsilon[E[F[\alpha B\nabla_2$	aw
(i) $C' \mapsto C', [G'$	(h) $\Upsilon[E[F[\alpha BC'$	aw
(j) $G' \mapsto G', [H'$	(i) $\Upsilon[E[F[\alpha BC'[G'$	aw
(k) $H' \xrightarrow{a} H$	(j) $\Upsilon[E[F[\alpha BC'[G'[H'$	aw
	(k) $\Upsilon[E[F[\alpha BC'[G'[H$	w

Tabla 6.3: Normalización de una transición compleja de un EPDA (izquierda) y emulación de la misma (derecha)

6.4. Equivalencia entre autómatas a pila embebidos sin estados y con estados

Para establecer la equivalencia entre la versión con estados y la versión sin estados del EPDA haremos uso de las transiciones complejas para EPDA sin estados definidas en el teorema 6.1 y de la forma normal para los EPDA con estados definida por Vijay-Shanker en [206]. Esta última establece que las transiciones deberán tener la forma

$$(q', [Z'_k \dots [Z'_{i+1}, Z_m \dots Z_1, [Z'_i \dots [Z'_1] \in \delta(q, a, Z)$$

donde $q, q' \in Q$, $a \in V_T \cup \{\epsilon\}$, $Z, Z_1, \dots, Z_m, Z'_1, \dots, Z'_k \in V_S$ y $m \leq 2$. En consecuencia, las transiciones podrán apilar un elemento, cambiar el elemento de la cima o sacar el elemento de la cima de la pila que ocupa la cima de la pila principal, y podrán crear nuevas pilas con un único elemento encima y/o debajo de dicha pila.

Teorema 6.2 *Para todo EPDA sin estados \mathcal{A} , existe un EPDA con estados \mathcal{A}' tal que el lenguaje aceptado por \mathcal{A} es igual al lenguaje aceptado por \mathcal{A}' .*

Demostración:

Sea $\mathcal{A} = (V_T, V_S, \Theta, \$_0, \$_f)$ un EPDA sin estados. El EPDA con estados $\mathcal{A}' = (Q, V_T, V_S, \delta, q, \emptyset, \$_0)$ acepta el mismo lenguaje (por pila vacía) que \mathcal{A} si las transiciones en δ se obtienen mediante una traducción adecuada de las transiciones en Θ . Consideremos todos los posibles tipos de transiciones en un EPDA sin estados:

SWAP: una transición $C \xrightarrow{a} F$ se traduce por una transición $(q, \epsilon, F, \epsilon) \in \delta(q, a, C)$.

PUSH: una transición $C \xrightarrow{a} CF$ se traduce por una transición $(q, \epsilon, CF, \epsilon) \in \delta(q, a, C)$.

POP: una transición $CF \xrightarrow{a} G$ se traduce por una transición $(q', \epsilon, \epsilon, \epsilon) \in \delta(q, a, F)$ más la transición $(q, \epsilon, G, \epsilon) \in \delta(q', \epsilon, C)$, donde q' es un estado que sólo se utiliza en estas dos transiciones.

WRAP-A: una transición $C \xrightarrow{a} C, [F$ se traduce por una transición $(q, \epsilon, C, [F) \in \delta(q, a, C)$.

WRAP-B: una transición $C \xrightarrow{a} [C, F$ se traduce por una transición $(q, [C, F, \epsilon) \in \delta(q, a, C)$.

UNWRAP: una transición $C, [F \xrightarrow{a} G$ se traduce por una transición $(q'', \epsilon, \epsilon, \epsilon) \in \delta(q, a, F)$ más una transición $(q'', \epsilon, G, \epsilon) \in \delta(q, \epsilon, C)$, donde q'' es un estado que sólo se utiliza en estas dos transiciones. Es importante recordar que las pilas vacías son eliminadas automáticamente de la cima en los EPDA con estados.

Adicionalmente, deberemos considerar las dos transiciones siguientes, que vacían la pila cuando se alcanza la configuración final ($[\$_0[\$_f, \epsilon)$:

$$(q', \epsilon, \epsilon, \epsilon) \in \delta(q, \epsilon, \$_f)$$

$$(q, \epsilon, \epsilon, \epsilon) \in \delta(q', \epsilon, \$_0)$$

El conjunto Q estará formado por el estado q y todos los estados q' y q'' utilizados en la traducción de transiciones POP y UNWRAP. \square

Teorema 6.3 *Para todo EPDA con estados \mathcal{A} , existe un EPDA sin estados \mathcal{A}' tal que el lenguaje aceptado por \mathcal{A} es igual al lenguaje aceptado por \mathcal{A}' .*

Demostración:

Dado un EPDA con estados $\mathcal{A} = (Q, V_T, V_S, \delta, q_0, Q_F, \$_0)$ construiremos un EPDA sin estados $\mathcal{A}' = (V_T, V'_S, \Theta, \$'_0, \$'_f)$ que acepte el mismo lenguaje que es aceptado por \mathcal{A} por pila vacía. El conjunto V'_S estará formado por pares $\langle Z, q \rangle$ y $\langle Z^0, q \rangle$, donde $Z \in V_S \cup \{-\}$ y $q \in Q \cup \{-\}$, y por los elementos inicial $\$'_0 = \langle \$_0, - \rangle$ y final $\$'_f = \langle -, - \rangle$.

Las transiciones en Θ tendrán el formato de las transiciones descritas en el teorema 6.1 y serán el resultado de traducir las transiciones en δ . Supondremos que las transiciones de \mathcal{A} están en la forma normal definida por Vijay-Shanker en [206]. Consideremos cada uno de los posibles casos:

- Una transición

$$(q', [F_k \dots [F_{i+1}, Z_1, [F_i \dots [F_1) \in \delta(q, a, Z)$$

se traduce por una transición

$$\langle Z, q \rangle \xrightarrow{a} [\langle F_k^0, q' \rangle \dots [\langle F_{i+1}^0, q' \rangle, \langle Z_1, q' \rangle, [\langle F_i^0, q' \rangle \dots [\langle F_1^0, q' \rangle$$

- Una transición

$$(q', [F_k \dots [F_{i+1}, Z_2 Z_1, [F_i \dots [F_1) \in \delta(q, a, Z)$$

se traduce por una transición

$$\langle Z, q \rangle \xrightarrow{a} [\langle F_k^0, q' \rangle \dots [\langle F_{i+1}^0, q' \rangle, \langle Z_2, q' \rangle \langle Z_1, q' \rangle, [\langle F_i^0, q' \rangle \dots [\langle F_1^0, q' \rangle$$

- Una transición

$$(q', [F_k \dots [F_{i+1}, \epsilon, [F_i \dots [F_1) \in \delta(q, a, Z)$$

se traduce por un conjunto de transiciones

$$\langle D, q'' \rangle \langle Z, q \rangle \xrightarrow{a} [\langle F_k^0, q' \rangle \dots [\langle F_{i+1}^0, q' \rangle, \langle D, q' \rangle, [\langle F_i^0, q' \rangle \dots [\langle F_1^0, q' \rangle$$

para todo $\langle D, q'' \rangle \in V'_S$.

Adicionalmente, incluiremos una transición

$$\langle F^0, q \rangle \mapsto \langle F^0, q \rangle \langle F, q \rangle$$

por todo $\langle F^0, q \rangle \in V'_S$ tal que $F \in V_S$ y $q \in Q$. Observamos que los pares $\langle F^0, q \rangle$ representan el papel de elemento inicial en cada una de las pilas individuales que componen la pila principal. Dicho elemento es necesario puesto que los EPDA con estados detectan si la pila en la cima de la pila principal está vacía, en cuyo caso la eliminan, todo ello sin intervención de transición alguna. Este comportamiento debe ser definido explícitamente en un EPDA sin estados mediante la creación de transiciones UNWRAP. La utilización de elementos $\langle F^0, q \rangle$ permite hacer corresponder una configuración $(q, \Upsilon[w])$ de un EPDA con estados con una configuración $(\Upsilon[\langle F^0, q \rangle], w)$ de un EPDA sin estados. En consecuencia, es necesario añadir a Θ una transición UNWRAP

$$\langle D, q' \rangle, [\langle F^0, q \rangle \mapsto \langle D, q \rangle$$

por cada par

$$\begin{aligned} \langle D, q' \rangle &\in V'_S - \{\langle \$_0, - \rangle\} \\ \langle F^0, q \rangle &\in V'_S \end{aligned}$$

Las dos transiciones siguientes

$$\begin{aligned} \langle \$_0, - \rangle &\mapsto \langle \$_0, - \rangle [\langle \$_0^0, - \rangle \\ \langle \$_0^0, - \rangle &\mapsto \langle \$_0^0, - \rangle \langle \$_0, q_0 \rangle \end{aligned}$$

configuran la pila inicial $\langle \$_0, - \rangle [\langle \$_0^0, - \rangle \langle \$_0, q_0 \rangle$ sobre la cual pueden comenzar a ser aplicadas las transiciones definidas anteriormente.

Una cadena w es reconocida por \mathcal{A} cuando se obtiene una configuración (q, ϵ, ϵ) . Dicha configuración tiene su equivalente en la configuración $(\langle \$_0, - \rangle [\langle \$_0^0, q \rangle], \epsilon)$ de \mathcal{A}' . Las transiciones

$$\langle \$_0^0, q \rangle \mapsto \langle -, - \rangle$$

para todo $q \in Q$ permiten obtener la configuración $(\langle \$_0, - \rangle [\langle -, - \rangle], \epsilon)$, que es la configuración final de \mathcal{A}' . \square

Corolario 6.4 *Los EPDA sin estados son equivalentes a los EPDA con estados.*

Demostración:

El resultado enunciado se obtiene directamente a partir de los teoremas 6.2 y 6.3. \square

6.5. Esquemas de compilación de gramáticas independientes del contexto

Un modo sencillo de realizar un esquema de compilación para gramáticas independientes del contexto en autómatas a pila embebidos consiste en emular el comportamiento de un autómata a pila mediante un EPDA.

Partiendo de la configuración inicial $([\$_0], w)$ de un EPDA, podríamos pensar en realizar la emulación de un autómata a pila mediante la aplicación de las transiciones SWAP, PUSH y POP de este último sobre la pila $[\$_0]$. Sin embargo, este enfoque impide alcanzar una configuración final del EPDA, puesto que estas tienen la forma $([\$_0] \$_f, \epsilon)$, con dos pilas unitarias almacenadas en la pila principal.

Resolvemos el problema estableciendo que una configuración $(\$_0 B_1 B_2 \dots B_n, w)$ de un autómata a pila equivale a una configuración $([\$_0 [B_1 [B_1 \dots [B_1, w)$ de un EPDA, con lo cual las pilas unitarias pasan a representar el papel de los símbolos de pila. Con respecto a las transiciones, tenemos que:

- Las transiciones SWAP $C \mapsto F$ permanecen sin cambios.
- Las transiciones PUSH $C \mapsto CF$ del autómata a pila son reemplazadas por transiciones WRAP-A $C \mapsto C, [F$ en el autómata a pila embebido.
- Las transiciones POP $CF \mapsto G$ del autómata a pila son reemplazadas por transiciones UNWRAP $C, [F \mapsto G$ en el autómata a pila embebido.

El caso inverso también es válido, de tal modo que un autómata a pila embebido cuyas configuraciones sólo contienen pilas unitarias y que sólo utiliza transiciones SWAP, WRAP-A y UNWRAP es equivalente a un autómata a pila cuyas configuraciones se obtienen a partir de las configuraciones del EPDA eliminando los símbolos $[$. El conjunto de transiciones contendrá las transiciones SWAP y el resultado de convertir las transiciones WRAP-A en transiciones PUSH y las transiciones UNWRAP en transiciones POP.

A continuación mostramos un esquema de compilación genérico de gramáticas independientes del contexto en autómatas a pila embebidos, obtenido a partir del esquema de compilación 5.1 mediante la aplicación de las transformaciones mencionadas anteriormente.

Esquema de compilación 6.1 El esquema de compilación genérico de una gramática independiente del contexto en un autómata a pila embebido queda definido por el siguiente conjunto de reglas y los elementos inicial $\$_0$ y final \overleftarrow{S} .

$$\begin{array}{ll}
\text{[INIT]} & \$_0 \mapsto \$_0 \quad [\nabla_{0,0} \\
\text{[CALL]} & \nabla_{r,s} \mapsto \nabla_{r,s} \quad [\overrightarrow{A_{r,s+1}} \\
\text{[SEL]} & \overrightarrow{A_{r,0}} \mapsto \nabla_{r,0} \quad r \neq 0 \\
\text{[PUB]} & \nabla_{r,n_r} \mapsto \overleftarrow{A_{r,0}} \\
\text{[RET]} & \nabla_{r,s} \quad [\overleftarrow{A_{r,s+1}} \mapsto \nabla_{r,s+1} \\
\text{[SCAN]} & \overrightarrow{A_{r,0}} \xrightarrow{a} \overleftarrow{A_{r,0}} \quad A_{r,0} \rightarrow a
\end{array}$$

§

6.6. Esquemas de compilación de gramáticas de adjunción de árboles

Un esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido debe realizar las tareas siguientes:

- Recorrer los nodos de los árboles elementales.
- En el caso de la adjunción del árbol auxiliar β en un nodo N^γ :
 - Al llegar al nodo N^γ , suspender el recorrido de γ y comenzar el recorrido de β a partir de su raíz.

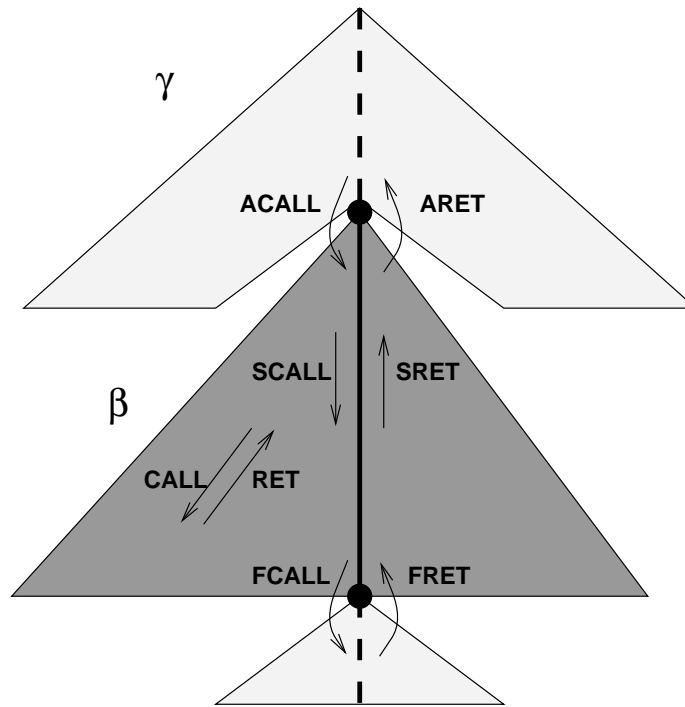


Figura 6.6: Reglas de compilación para TAG

- Al alcanzar el nodo pie de β , suspender el recorrido de este último para continuar el recorrido del subárbol de γ que cuelga del nodo N^γ .

El recorrido de un árbol elemental es equivalente al recorrido del conjunto de producciones independientes del contexto que lo componen, por lo que podemos utilizar para esta tarea las reglas de compilación **[CALL]**, **[SEL]**, **[PUB]** y **[RET]** definidas en el esquema de compilación 6.1. Al igual que en el capítulo 3, consideraremos una producción adicional $\top^\beta \rightarrow \mathbf{R}^\alpha$ para cada árbol inicial α y dos producciones adicionales para cada árbol auxiliar β : $\top^\beta \rightarrow \mathbf{R}^\beta$ y $\mathbf{F}^\beta \rightarrow \perp^\beta$, donde \mathbf{R}^β y \mathbf{F}^β se refieren a los nodos raíz y pie de β , respectivamente.

El tratamiento de la adjunción implica suspender el recorrido de un árbol al llegar a la raíz del árbol auxiliar, para retomarlo más tarde al alcanzar el nodo pie de dicho árbol auxiliar. Para ello hemos de poder transmitir, a través de la espina del árbol auxiliar, el nodo en el cual se realizó la adjunción. En el caso de que se realicen adjunciones en nodos de la espina, dichas adjunciones se irán apilando. Utilizaremos las propias pilas del EPDA para almacenar las pilas de adjunciones pendientes. Para ello, dotaremos de la siguiente semántica a las pilas del EPDA: dada una pila $[\alpha B$, el elemento B de la cima nos informa del punto en el que se encuentra el recorrido de un árbol elemental mientras que la parte restante α contiene la pila de adjunciones pendientes en dicho punto. La figura 6.6 muestra de modo intuitivo la misión de cada una de las reglas de compilación:

- La regla de compilación **[SCALL]** será la encargada de crear las transiciones que propaguen la pila de adjunciones pendientes a través de la espina de los árboles auxiliares, desde la raíz al pie.
- La regla **[SRET]** es una regla **[RET]** que trata con elementos de la espina de un árbol auxiliar. Se define con el fin de formar un par **[SCALL]**–**[SRET]** análogo al par **[CALL]**–**[RET]**.

- Al llegar al nodo de adjunción N^γ , deberemos apilar dicho nodo en la pila de adjunciones pendientes y pasar a la raíz del árbol auxiliar β . La regla de compilación [ACALL] se encargará de crear las transiciones que realicen estas tareas.
- La regla de compilación [ARET] es un tipo especial de regla [RET] que permite continuar el recorrido de γ una vez que ha terminado la adjunción de β .
- Al llegar al nodo pie de β , deberemos eliminar el nodo N^γ de la pila de adjunciones pendientes y continuar el recorrido en el árbol que cuelga de dicho nodo. De crear las transiciones que realicen estas tareas se encargará la regla de compilación [FCALL].
- La regla de compilación [FRET] es un tipo especial de regla [RET] que permite continuar el recorrido a partir del nodo pie de β .

A continuación definimos un esquema de compilación genérico, derivado del esquema de compilación 6.1, en el cual se ha parametrizado la información concerniente al recorrido de los árboles elementales.

Esquema de compilación 6.2 El esquema de compilación genérico de una gramática de adjunción de árboles en un autómata a pila embebido queda definido por el conjunto de reglas mostrado en la figura 6.4 y los elementos inicial $\$0$ y final \overline{S} . Es interesante señalar que las pilas de adjunciones pendientes no almacenan directamente los nodos $N_{r,s+1}^\gamma$ en los que se realizaron las adjunciones sino el elemento $\nabla_{r,s}^\gamma$ que indica el punto en el que se lanzó la adjunción, almacenado en las pilas bajo la forma $\Delta_{r,s}^\gamma$ para evitar confusiones. Podemos ver un símbolo Δ como un símbolo ∇ en espera de la finalización de una adjunción.

Las reglas de compilación [ACALL] y [FCALL] podrían escribirse alternativamente de la siguiente forma:

$$\begin{aligned} \text{[ACALL]} \quad \nabla_{r,s}^\gamma &\mapsto [\nabla_{r,s}^\gamma, \Delta_{r,s}^\gamma, \overline{\top}^\beta] & \beta \in \text{adj}(N_{r,s+1}^\gamma) \\ \text{[FCALL]} \quad \Delta_{r,s}^\gamma, \nabla_{f,0}^\beta &\mapsto [\nabla_{f,0}^\beta, \overline{N_{r,s+1}^\gamma}] & N_{f,0}^\beta = \mathbf{F}^\beta, \quad \beta \in \text{adj}(N_{r,s+1}^\gamma) \end{aligned}$$

pero como las transiciones resultantes no son transiciones elementales, hemos tenido que descomponer dichas reglas de compilación en dos pares de reglas: [ACALL-a] más [ACALL-b] y [FCALL-a] más [FCALL-b]. §

El esquema de compilación genérico establece que las adjunciones se reconocen de modo descendente, puesto que al pasar a la raíz de un árbol auxiliar se apila el nodo de adjunción en la pila de adjunciones pendientes y al llegar al nodo pie de dicho árbol se saca de la pila de adjunciones pendientes.

Es interesante remarcar que la regla de compilación [CALL] utiliza una transición WRAP-A con el significado de “crear una nueva pila que indique un nodo del árbol γ con una pila vacía de adjunciones pendientes”, mientras que la regla de compilación [SCALL] utiliza una transición WRAP-B a la que dota del significado “Crear una nueva pila que indique el nodo del árbol γ y pasarle la pila de adjunciones pendientes”. En la figura 6.4 se observa cómo la parte sombreada (en nuestro caso, la pila de adjunciones pendientes) permanece en su posición original en la operación WRAP-A mientras que en la operación WRAP-B la parte sombreada es pasada a la nueva pila en la cima, al tiempo que la pila que queda debajo sólo conserva el elemento C de su cima. Cuando se continúa el recorrido del árbol auxiliar a partir del pie ya no queda rastro de

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}^\alpha$	$\alpha \in \mathbf{I}$
[CALL]	$\nabla_{r,s}^\gamma \mapsto \nabla_{r,s}^\gamma [\overrightarrow{N_{r,s+1}^\gamma}$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SCALL]	$\nabla_{r,s}^\beta \mapsto [\nabla_{r,s}^\beta, \overrightarrow{N_{r,s+1}^\beta}$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SEL]	$\overrightarrow{N_{r,0}^\gamma} \mapsto \nabla_{r,0}^\gamma$	
[PUB]	$\nabla_{r,nr}^\gamma \mapsto \overleftarrow{N_{r,0}^\gamma}$	
[RET]	$\nabla_{r,s}^\gamma, [\overleftarrow{N_{r,s+1}^\gamma} \mapsto \nabla_{r,s+1}^\gamma$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SRET]	$\nabla_{r,s}^\beta, [\overleftarrow{N_{r,s+1}^\beta} \mapsto \nabla_{r,s+1}^\beta$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SCAN]	$\overrightarrow{N_{r,0}^\gamma} \xrightarrow{a} \overleftarrow{N_{r,0}^\gamma}$	$N_{r,0}^\gamma \rightarrow a$
[ACALL-a]	$\nabla_{r,s}^\gamma \mapsto [\nabla_{r,s}^\gamma, \Delta_{r,s}^\gamma$	$\text{adj}(N_{r,s+1}^\gamma) \neq \{\mathbf{nil}\}$
[ACALL-b]	$\Delta_{r,s}^\gamma \mapsto \Delta_{r,s}^\gamma \overleftarrow{\top}^\beta$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[ARET]	$\nabla_{r,s}^\gamma, [\overleftarrow{\top}^\beta \mapsto \nabla_{r,s+1}^\gamma$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[FCALL-a]	$\nabla_{f,0}^\beta \mapsto [\nabla_{f,0}^\beta, \perp$	$N_{f,0}^\beta = \mathbf{F}^\beta$
[FCALL-b]	$\Delta_{r,s}^\gamma \perp \mapsto \overrightarrow{N_{r,s+1}^\gamma}$	
[FRET]	$\nabla_{f,0}^\beta, [\overleftarrow{N_{r,s+1}^\gamma} \mapsto \nabla_{f,1}^\beta$	$N_{f,0}^\beta = \mathbf{F}^\beta, \beta \in \text{adj}(N_{r,s+1}^\gamma)$

Tabla 6.4: Reglas del esquema de compilación genérico de TAG en EPDA

la pila de adjunciones pendientes, lo que permite utilizar transiciones UNWRAP en las reglas [RET] y [SRET].

El esquema de compilación genérico no establece ninguna restricción sobre la estrategia utilizada para recorrer los árboles elementales. A continuación se definen, de acuerdo con la tabla 5.2, los esquemas de compilación correspondientes a tres estrategias particulares aplicadas al recorrido de los árboles elementales: ascendente, Earley y descendente.

6.6.1. Estrategia descendente

Esquema de compilación 6.3 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido que incorpora una estrategia descendente para el recorrido de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 6.5 y los elementos inicial $\$0$ y final \square . §

6.6.2. Estrategia Earley

Esquema de compilación 6.4 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido que incorpora una estrategia Earley para el recorrido

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}^\alpha$	$\alpha \in \mathbf{I}$
[CALL]	$\nabla_{r,s}^\gamma \mapsto \nabla_{r,s}^\gamma, [N_{r,s+1}^\gamma$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SCALL]	$\nabla_{r,s}^\beta \mapsto [\nabla_{r,s}^\beta, N_{r,s+1}^\beta$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SEL]	$N_{r,0}^\gamma \mapsto \nabla_{r,0}^\gamma$	
[PUB]	$\nabla_{r,n_r}^\gamma \mapsto \square$	
[RET]	$\nabla_{r,s}^\gamma, [\square \mapsto \nabla_{r,s+1}^\gamma$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SRET]	$\nabla_{r,s}^\beta, [\square \mapsto \nabla_{r,s+1}^\beta$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SCAN]	$N_{r,0}^\gamma \xrightarrow{a} \square$	$N_{r,0}^\gamma \rightarrow a$
[ACALL-a]	$\nabla_{r,s}^\gamma \mapsto [\nabla_{r,s}^\gamma, \Delta_{r,s}^\gamma$	$\text{adj}(N_{r,s+1}^\gamma) \neq \{\mathbf{nil}\}$
[ACALL-b]	$\Delta_{r,s}^\gamma \mapsto \Delta_{r,s}^\gamma \top^\beta$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[ARET]	$\nabla_{r,s}^\gamma, [\square \mapsto \nabla_{r,s+1}^\gamma$	
[FCALL-a]	$\nabla_{f,0}^\beta \mapsto [\nabla_{f,0}^\beta, \perp$	$N_{f,0}^\beta = \mathbf{F}^\beta$
[FCALL-b]	$\Delta_{r,s}^\gamma \perp \mapsto N_{r,s+1}^\gamma$	
[FRET]	$\nabla_{f,0}^\beta, [\square \mapsto \nabla_{f,1}^\beta$	$N_{f,0}^\beta = \mathbf{F}^\beta, \beta \in \text{adj}(N_{r,s+1}^\gamma)$

Tabla 6.5: Reglas del esquema de compilación descendente de TAG en EPDA

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}^\alpha$	$\alpha \in \mathbf{I}$
[CALL]	$\nabla_{r,s}^\gamma \mapsto \nabla_{r,s}^\gamma, [\overline{N_{r,s+1}^\gamma}$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SCALL]	$\nabla_{r,s}^\beta \mapsto [\nabla_{r,s}^\beta, \overline{N_{r,s+1}^\beta}$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SEL]	$\overline{N_{r,0}^\gamma} \mapsto \nabla_{r,0}^\gamma$	
[PUB]	$\nabla_{r,n_r}^\gamma \mapsto \overline{\overline{N_{r,0}^\gamma}}$	
[RET]	$\nabla_{r,s}^\gamma, [\overline{\overline{N_{r,s+1}^\gamma}} \mapsto \nabla_{r,s+1}^\gamma$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SRET]	$\nabla_{r,s}^\beta, [\overline{\overline{N_{r,s+1}^\beta}} \mapsto \nabla_{r,s+1}^\beta$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SCAN]	$\overline{N_{r,0}^\gamma} \xrightarrow{a} \overline{\overline{N_{r,0}^\gamma}}$	$N_{r,0}^\gamma \rightarrow a$
[ACALL-a]	$\nabla_{r,s}^\gamma \mapsto [\nabla_{r,s}^\gamma, \Delta_{r,s}^\gamma$	$\text{adj}(N_{r,s+1}^\gamma) \neq \{\mathbf{nil}\}$
[ACALL-b]	$\Delta_{r,s}^\gamma \mapsto \Delta_{r,s}^\gamma \overline{\top}^\beta$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[ARET]	$\nabla_{r,s}^\gamma, [\overline{\overline{\top}^\beta} \mapsto \nabla_{r,s+1}^\gamma$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[FCALL-a]	$\nabla_{f,0}^\beta \mapsto [\nabla_{f,0}^\beta, \perp$	$N_{f,0}^\beta = \mathbf{F}^\beta$
[FCALL-b]	$\Delta_{r,s}^\gamma \perp \mapsto \overline{\overline{N_{r,s+1}^\gamma}}$	
[FRET]	$\nabla_{f,0}^\beta, [\overline{\overline{N_{r,s+1}^\gamma}} \mapsto \nabla_{f,1}^\beta$	$N_{f,0}^\beta = \mathbf{F}^\beta, \beta \in \text{adj}(N_{r,s+1}^\gamma)$

Tabla 6.6: Reglas del esquema de compilación Earley de TAG en EPDA

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}^\alpha$	$\alpha \in I$
[CALL]	$\nabla_{r,s}^\gamma \mapsto \nabla_{r,s}^\gamma, [\square$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SCALL]	$\nabla_{r,s}^\beta \mapsto [\nabla_{r,s}^\beta, \square$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SEL]	$\square \mapsto \nabla_{r,0}^\gamma$	
[PUB]	$\nabla_{r,n_r}^\gamma \mapsto N_{r,0}^\gamma$	
[RET]	$\nabla_{r,s}^\gamma, [N_{r,s+1}^\gamma \mapsto \nabla_{r,s+1}^\gamma$	$N_{r,s+1}^\gamma \notin \text{espina}(\gamma), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\gamma)$
[SRET]	$\nabla_{r,s}^\beta, [N_{r,s+1}^\beta \mapsto \nabla_{r,s+1}^\beta$	$N_{r,s+1}^\beta \in \text{espina}(\beta), \mathbf{nil} \in \text{adj}(N_{r,s+1}^\beta)$
[SCAN]	$\square \xrightarrow{a} N_{r,0}^\gamma$	$N_{r,0}^\gamma \rightarrow a$
[ACALL-a]	$\nabla_{r,s}^\gamma \mapsto [\nabla_{r,s}^\gamma, \Delta_{r,s}^\gamma$	$\text{adj}(N_{r,s+1}^\gamma) \neq \{\mathbf{nil}\}$
[ACALL-b]	$\Delta_{r,s}^\gamma \mapsto \Delta_{r,s}^\gamma, \square$	
[ARET]	$\nabla_{r,s}^\gamma, [\top^\beta \mapsto \nabla_{r,s+1}^\gamma$	$\beta \in \text{adj}(N_{r,s+1}^\gamma)$
[FCALL-a]	$\nabla_{f,0}^\beta \mapsto [\nabla_{f,0}^\beta, \perp$	$N_{f,0}^\beta = \mathbf{F}^\beta$
[FCALL-b]	$\Delta_{r,s}^\gamma, \perp \mapsto \square$	
[FRET]	$\nabla_{f,0}^\beta, [N_{r,s+1}^\gamma \mapsto \nabla_{f,1}^\beta$	$N_{f,0}^\beta = \mathbf{F}^\beta, \beta \in \text{adj}(N_{r,s+1}^\gamma)$

Tabla 6.7: Reglas del esquema de compilación ascendente de TAG en EPDA

de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 6.6 y los elementos inicial $\$0$ y final \overline{S} . §

6.6.3. Estrategia ascendente

Esquema de compilación 6.5 El esquema de compilación de una gramática de adjunción de árboles en un autómata a pila embebido que incorpora una estrategia ascendente para el recorrido de los árboles elementales queda definido por el conjunto de reglas mostrado en la tabla 6.7 y los elementos inicial $\$0$ y final S . §

6.7. Esquemas de compilación de gramáticas lineales de índices

Para el análisis de gramáticas lineales de índices mediante autómatas a pila embebidos utilizaremos la semántica dada por Vijay-Shanker en [206] a las pilas del EPDA: una pila $[\alpha B$ se corresponde con el símbolo $B[\alpha]$ de una gramática lineal de índices. En consecuencia, para emular una derivación de una gramática lineal de índices en un EPDA es preciso ir modificando las pilas del autómata de acuerdo con los cambios en las pilas de índices indicados por la gramática. De ello se encargan las reglas de compilación descritas en la tabla 6.8.

La única novedad con respecto a las reglas utilizadas en el esquema de compilación 6.1 para gramáticas independientes del contexto radica en la introducción de la regla de compilación

Regla	Tarea
[INIT]	Inicia los cálculos a partir de la pila inicial.
[CALL]	Requiere el análisis de un determinado elemento gramatical $A[]$ que no es un hijo dependiente, lo cual implica situar en la cima de la pila principal una nueva pila $[A$.
[SCALL]	Requiere el análisis de un determinado elemento gramatical que es un hijo dependiente, lo que implica tener que pasarle la pila de índices con los cambios correspondientes.
[SEL]	Selecciona una producción.
[PUB]	Determina que una producción de la gramática ha sido completamente analizada.
[RET]	Continúa el proceso de análisis después de que se haya completado una producción, lo cual implica eliminar una pila $[A$ de la cima de la pila principal.
[SCAN]	Reconoce los terminales que componen la cadena de entrada.

Tabla 6.8: Tipos de reglas de los esquemas de compilación de LIG en EPDA

[SCALL] utilizada para manejar las pilas de índices. Debido a las características propias de los autómatas a pila embebidos, las pilas de índices asociadas a los terminales se calculan de modo descendente, de tal forma que dada una producción $A \rightarrow \Upsilon_1 B \Upsilon_2$, la pila de índices asociada a B se calcula a partir de la pila asociada a A en el momento de comenzar a analizar B . Cuando se termine de analizar B , su pila asociada será vacía. Quiere esto decir que todas las comprobaciones acerca de la buena formación de las pilas se chequean en la fase descendente del algoritmo y que durante la fase ascendente todas las pilas de índices que se propagan son pilas vacías.

Con respecto al tratamiento de los no-terminales de la gramática lineal de índices, los autómatas a pila embebidos no imponen ninguna restricción, circunstancia que aprovecharemos para definir un esquema de compilación para una estrategia genérica en la cual se ha parametrizado la información predicha en la fase descendente y la información propagada en la fase ascendente. Posteriormente definiremos los esquemas correspondientes a las estrategias descendente, Earley y ascendente.

Esquema de compilación 6.6 El esquema de compilación genérico de una gramática lineal de índices en un autómata a pila embebido queda definido por el conjunto de reglas mostrado en la tabla 6.9 y los elementos inicial $\$_0$ y final \overline{S} . Las reglas de compilación [SCALL-2] y [SCALL-3] podrían escribirse alternativamente de la forma:

$$\text{[SCALL-2]} \quad \nabla_{r,s} \mapsto [\nabla_{r,s}, \gamma' \overrightarrow{A_{r,s+1}} \quad A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma']\Upsilon_2$$

$$\text{[SCALL-3]} \quad \gamma \nabla_{r,s} \mapsto [\nabla_{r,s}, \overrightarrow{A_{r,s+1}} \quad A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ]\Upsilon_2$$

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \mapsto \nabla_{r,s}, [\overrightarrow{A_{r,s+1}}$	$A_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} [] \Upsilon_2$
[SCALL-1]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \overrightarrow{A_{r,s+1}}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-2a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \langle \gamma', r, s+1 \rangle$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma'] \Upsilon_2$
[SCALL-2b]	$\langle \gamma', r, s+1 \rangle \mapsto \langle \gamma', r, s+1 \rangle \overrightarrow{A_{r,s+1}}$	
[SCALL-3a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \nabla'_{r,s}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-3b]	$\langle \gamma, t, u \rangle \nabla'_{r,s} \mapsto \overrightarrow{A_{r,s+1}}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SEL]	$\overrightarrow{A_{r,0}} \mapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \mapsto \overleftarrow{A_{r,0}}$	
[RET]	$\nabla_{r,s}, [\overleftarrow{A_{r,s+1}} \mapsto \nabla_{r,s+1}$	$A_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} \Upsilon_2$
[SCAN]	$\overrightarrow{A_{r,0}} \xrightarrow{a} \overleftarrow{A_{r,0}}$	$A_{r,0}[] \rightarrow a$

Tabla 6.9: Reglas del esquema de compilación genérico de LIG en EPDA

pero en tal caso las producciones involucradas no formarían parte de la familia de transiciones elementales de los EPDA. Esta circunstancia nos ha llevado a descomponer la regla de compilación [SCALL-2] en dos reglas [SCALL-2a] y [SCALL-2b] y a descomponer la regla [SCALL-3] en dos reglas [SCALL-3a] y [SCALL-3b]. Como un efecto colateral, las pilas del autómata tendrán la forma $[\alpha B$, donde B será un no-terminal de la gramática lineal de índices y α estará formado por una sucesión de triples $\langle \gamma, r, s \rangle$, donde γ es un índices mientras que r y s señalan una posición s en una producción r . La proyección del primer componente de dichos triples proporciona la pila de índices asociada a B . Los componentes r y s sólo se utilizan en la regla de compilación [SCALL-2b] mientras que son ignorados en [SCALL-3b]. §

6.7.1. Estrategia descendente

Esquema de compilación 6.7 El esquema de compilación con estrategia descendente de una gramática lineal de índices en un autómata a pila embebido queda definido por el conjunto de reglas de la tabla 6.10 y los elementos inicial $\$0$ y final \square . §

6.7.2. Estrategia Earley

Esquema de compilación 6.8 El esquema de compilación con estrategia Earley de una gramática lineal de índices en un autómata a pila embebido queda definido por el conjunto de reglas de la tabla 6.11 y los elementos inicial $\$0$ y final $\overline{\square}$. §

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \mapsto \nabla_{r,s}, [A_{r,s+1}$	$\mathbf{A}_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} [] \Upsilon_2$
[SCALL-1]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, A_{r,s+1}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-2a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \langle \gamma', r, s+1 \rangle$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma'] \Upsilon_2$
[SCALL-2b]	$\langle \gamma', r, s+1 \rangle \mapsto \langle \gamma', r, s+1 \rangle A_{r,s+1}$	
[SCALL-3a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \nabla'_{r,s}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-3b]	$\langle \gamma, t, u \rangle \nabla'_{r,s} \mapsto A_{r,s+1}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SEL]	$A_{r,0} \mapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \mapsto \square$	
[RET]	$\nabla_{r,s}, [\square \mapsto \nabla_{r,s+1}$	$\mathbf{A}_{r,0} \rightarrow \Upsilon_1 \mathbf{A}_{r,s+1} \Upsilon_2$
[SCAN]	$A_{r,0} \xrightarrow{a} \square$	$A_{r,0}[] \rightarrow a$

Tabla 6.10: Reglas del esquema de compilación descendente de LIG en EPDA

6.7.3. Estrategia ascendente

Esquema de compilación 6.9 El esquema de compilación con estrategia ascendente de una gramática lineal de índices en un autómata a pila embebido queda definido por el conjunto de reglas de la tabla 6.12 y los elementos inicial $\$0$ y final S . §

6.8. Lenguajes de adjunción de árboles y EPDA

Vijay-Shanker estableció en [206] que la clase de los lenguajes aceptados por los EPDA era equivalente a la clase de los lenguajes de adjunción de árboles. Para ello se valió de una técnica que permite obtener una gramática de núcleo a partir de un EPDA y viceversa. En esta sección ofrecemos un modo distinto de establecer la equivalencia entre los lenguajes aceptados por los EPDA y los lenguajes de adjunción de árboles. Para ello haremos uso de los esquemas de compilación definidos previamente.

Teorema 6.5 *Los lenguajes de adjunción de árboles son un subconjunto de los lenguajes aceptados por la clase de los autómatas a pila embebidos.*

Demostración:

Por el esquema de compilación 6.2, a partir de cualquier TAG es posible construir un EPDA que acepta el lenguaje reconocido por dicha gramática. Análogamente, por el esquema de compilación 6.6, a partir de cualquier LIG es posible construir un EPDA que acepta el lenguaje reconocido por dicha gramática. □

Teorema 6.6 *La clase de los lenguajes aceptados por los EPDA es un subconjunto de los lenguajes de adjunción de árboles.*

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \mapsto \nabla_{r,s}, [\overline{A_{r,s+1}}$	$A_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} [] \Upsilon_2$
[SCALL-1]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \overline{A_{r,s+1}}$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-2a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \langle \gamma', r, s+1 \rangle$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma'] \Upsilon_2$
[SCALL-2b]	$\langle \gamma', r, s+1 \rangle \mapsto \langle \gamma', r, s+1 \rangle \overline{A_{r,s+1}}$	
[SCALL-3a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \nabla'_{r,s}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-3b]	$\langle \gamma, t, u \rangle \nabla'_{r,s} \mapsto \overline{A_{r,s+1}}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SEL]	$\overline{A_{r,0}} \mapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \mapsto \overline{A_{r,0}}$	
[RET]	$\nabla_{r,s}, [\overline{A_{r,s+1}} \mapsto \nabla_{r,s+1}$	$A_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} \Upsilon_2$
[SCAN]	$\overline{A_{r,0}} \xrightarrow{a} \overline{A_{r,0}}$	$A_{r,0}[] \rightarrow a$

Tabla 6.11: Reglas del esquema de compilación Earley de LIG en EPDA

Demostración:

Mostraremos que para todo EPDA existe una gramática lineal de índices tal que el lenguaje reconocido por la gramática coincide con el lenguaje aceptado por el autómata.

Sea $\mathcal{A} = (V_T, V_S, \Theta, \$0, \$f)$ un autómata a pila embebido. Construiremos una gramática lineal de índices $\mathcal{L} = (V_T, V_N, V_I, S, P)$, donde $V_I = V_S$ y el conjunto V_N de no-terminales estará formado por pares $\langle A, B \rangle$ tales que $A, B \in V_S$. Para que \mathcal{L} reconozca el lenguaje aceptado por \mathcal{A} el conjunto de producciones en P ha de construirse a partir de las transiciones en Θ de la siguiente manera:

- Para toda transición $C \xrightarrow{a} F$ y para todo $E \in V_S$ creamos una producción
$$\langle C, E \rangle[\circ\circ] \rightarrow a \langle F, E \rangle[\circ\circ]$$
- Para toda transición $C \xrightarrow{a} CF$ y para todo $E \in V_S$ creamos una producción
$$\langle C, E \rangle[\circ\circ] \rightarrow a \langle F, E \rangle[\circ\circ C]$$
- Para toda transición $CF \xrightarrow{a} G$ y para todo $E \in V_S$ creamos una producción
$$\langle F, E \rangle[\circ\circ C] \rightarrow a \langle G, E \rangle[\circ\circ]$$
- Para todo par de transiciones $C, [F \xrightarrow{a} G$ y $C \xrightarrow{b} C, [F'$, y para todo $E \in V_S$ creamos una producción
$$\langle C, E \rangle[\circ\circ] \rightarrow b \langle F', F \rangle[] a \langle G, E \rangle[\circ\circ]$$
- Para todo par de transiciones $C, [F \xrightarrow{a} G$ y $C \xrightarrow{b} [C, F'$, y para todo $E \in V_S$ creamos una producción
$$\langle C, E \rangle[\circ\circ] \rightarrow b \langle F', F \rangle[\circ\circ] a \langle G, E \rangle[]$$
- Para todo $E \in V_S$ creamos una producción
$$\langle E, E \rangle[] \rightarrow \epsilon$$
- Para toda transición $\$0 \xrightarrow{a} \$0, [F$ o $\$0 \xrightarrow{a} [\$0, F$, donde $F \in V_S - \{\$0\}$, creamos una producción
$$\langle \$0, \$0 \rangle[\circ\circ] \rightarrow a \langle F, \$f \rangle[\circ\circ]$$

[INIT]	$\$0 \mapsto \$0, [\nabla_{0,0}$	
[CALL]	$\nabla_{r,s} \mapsto \nabla_{r,s}, [\square$	$\mathbf{A}_{r,0} \rightarrow \Upsilon_1 A_{r,s+1} [] \Upsilon_2$
[SCALL-1]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \square$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-2a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \langle \gamma', r, s+1 \rangle$	$A_{r,0}[\circ\circ] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ\gamma'] \Upsilon_2$
[SCALL-2b]	$\langle \gamma', r, s+1 \rangle \mapsto \langle \gamma', r, s+1 \rangle \square$	
[SCALL-3a]	$\nabla_{r,s} \mapsto [\nabla_{r,s}, \nabla'_{r,s}$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SCALL-3b]	$\langle \gamma, t, u \rangle \nabla'_{r,s} \mapsto \square$	$A_{r,0}[\circ\circ\gamma] \rightarrow \Upsilon_1 A_{r,s+1}[\circ\circ] \Upsilon_2$
[SEL]	$\square \mapsto \nabla_{r,0}$	$r \neq 0$
[PUB]	$\nabla_{r,n_r} \mapsto A_{r,0}$	
[RET]	$\nabla_{r,s}, [A_{r,s+1} \mapsto \nabla_{r,s+1}$	$\mathbf{A}_{r,0} \rightarrow \Upsilon_1 \mathbf{A}_{r,s+1} \Upsilon_2$
[SCAN]	$\square \xrightarrow{a} A_{r,0}$	$A_{r,0}[] \rightarrow a$

Tabla 6.12: Reglas del esquema de compilación ascendente de LIG en EPDA

- Para toda transición $B \xrightarrow{a} \$_f$ creamos una transición

$$\langle B, \$_f \rangle [] \rightarrow a$$

Con respecto al axioma de la gramática, tenemos que $S = \langle \$0, \$0 \rangle$.

Es posible mostrar que $\langle C, E \rangle [\alpha] \xrightarrow{*} w$ si y sólo si $([\alpha C, w] \vdash ([E, \epsilon])^*$, puesto que:

- Si una derivación $([\alpha C, w] \vdash ([E, \epsilon])^*$ es el resultado de aplicar la secuencia t_1, \dots, t_m de transiciones en Θ , entonces existe una secuencia p_1, \dots, p'_m de producciones en P tal que $\langle C, E \rangle [\alpha] \xrightarrow{*} w$ resultado de aplicar p_1, \dots, p'_m reconoce w . La demostración se realiza por inducción en la longitud de la derivación del autómata. El caso base lo constituye la derivación $([E, \epsilon]) \vdash ([E, \epsilon])^0$, para la que existe una transición $\langle E, E \rangle [] \rightarrow \epsilon$. Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación del autómata de longitud m . En el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que m :
 - Si $([\alpha C, aw] \vdash ([\alpha F, w] \vdash ([E, \epsilon], \exists \langle C, E \rangle [\circ\circ] \rightarrow a \langle F, E \rangle [\circ\circ]) \in P$, por hipótesis de inducción $\langle F, E \rangle [\alpha] \xrightarrow{*} w$ y en consecuencia $\langle C, E \rangle [\alpha] \xrightarrow{*} aw$.
 - Si $([\alpha C, aw] \vdash ([\alpha CF, w] \vdash ([E, \epsilon], \exists \langle C, E \rangle [\circ\circ] \rightarrow a \langle F, E \rangle [\circ\circ C]) \in P$, por hipótesis de inducción $\langle F, E \rangle [\alpha C] \xrightarrow{*} w$ y en consecuencia $\langle C, E \rangle [\alpha] \xrightarrow{*} aw$.
 - Si $([\alpha CF, aw] \vdash ([\alpha G, w] \vdash ([E, \epsilon], \exists \langle F, E \rangle [\circ\circ C] \rightarrow a \langle G, E \rangle [\circ\circ]) \in P$, por hipótesis de inducción $\langle G, E \rangle [\alpha] \xrightarrow{*} w$ y en consecuencia $\langle F, E \rangle [\alpha C] \xrightarrow{*} aw$.
 - Si $([\alpha C, bw_1aw_2] \vdash ([\alpha C[F', w_1aw_2] \vdash^{m_1} ([\alpha C[F, aw_2] \vdash ([\alpha G, w_2] \vdash^{m_2} ([E, \epsilon], \exists \langle C, E \rangle [\circ\circ] \rightarrow b \langle F', F \rangle [] a \langle G, E \rangle [\circ\circ]) \in P$, por hipótesis de inducción $\langle F', F \rangle [] \xrightarrow{*} w_1$ y $\langle G, E \rangle [\alpha] \xrightarrow{*} w_2$ y en consecuencia $\langle C, E \rangle [\alpha] \xrightarrow{*} aw_1bw_2$.
 - Si $([\alpha C, bw_1aw_2] \vdash ([C[\alpha F', w_1aw_2] \vdash^{m_1} ([C[F, aw_2] \vdash ([G, w_2] \vdash^{m_2} ([E, \epsilon], \exists \langle C, E \rangle [\circ\circ] \rightarrow b \langle F', F \rangle [\circ\circ] a \langle G, E \rangle [] \in P$, por hipótesis de inducción $\langle F', F \rangle [\alpha] \xrightarrow{*} w_1$ y $\langle G, E \rangle [] \xrightarrow{*} w_2$ y en consecuencia $\langle C, E \rangle [\alpha] \xrightarrow{*} aw_1bw_2$.
- Si una derivación izquierda $\langle C, E \rangle [\alpha] \xrightarrow{*} w$ reconoce la cadena w como resultado de aplicar la secuencia p_1, \dots, p_m de producciones en P , entonces existe una secuencia

de transiciones t_1, \dots, t'_m tal que la derivación $([\alpha C, w] \vdash^* ([E, \epsilon])$ es el resultado de aplicar la secuencia de transiciones t_1, \dots, t'_m . La demostración se realiza por inducción en la longitud de la derivación de la gramática. El caso base lo constituye la derivación $\langle E, E \rangle[\] \Rightarrow \epsilon$, para la que existe una derivación $([E, \epsilon] \vdash^0 ([E, \epsilon])$ en el autómata. Por hipótesis de inducción suponemos que la proposición se cumple para cualquier derivación del autómata de longitud m . En el paso de inducción verificamos que se cumple para cualquier posible derivación de longitud mayor que m :

- Si $\langle C, E \rangle[\alpha] \Rightarrow a \langle F, E \rangle[\alpha] \xrightarrow{m} aw$, existe una transición $C \xrightarrow{a} F$, por hipótesis de inducción $([\alpha F, w] \vdash^* ([E, \epsilon])$ y en consecuencia $([\alpha C, aw] \vdash^* ([E, \epsilon])$.
- Si $\langle C, E \rangle[\alpha] \Rightarrow a \langle F, E \rangle[\alpha C] \xrightarrow{m} aw$, existe una transición $C \xrightarrow{a} CF$, por hipótesis de inducción $([\alpha CF, w] \vdash^* ([E, \epsilon])$ y en consecuencia $([\alpha C, aw] \vdash^* ([E, \epsilon])$.
- Si $\langle F, E \rangle[\alpha C] \Rightarrow a \langle G, E \rangle[\alpha] \xrightarrow{m} aw$, existe una transición $CF \xrightarrow{a} G$, por hipótesis de inducción $([\alpha G, w] \vdash^* ([E, \epsilon])$ y en consecuencia $([\alpha CF, aw] \vdash^* ([E, \epsilon])$.
- Si $\langle C, E \rangle[\alpha] \Rightarrow b \langle F', F \rangle[\] a \langle G, E \rangle[\alpha] \xrightarrow{m_1} bw_1 a \langle G, E \rangle[\alpha] \xrightarrow{m_2} aw_1 bw_2$, existe un par de transiciones $C \xrightarrow{b} C, [F'$ y $C, [F \xrightarrow{a} G$, por hipótesis de inducción $([F', w_1] \vdash^* ([F, \epsilon])$ y $([\alpha G, w_2] \vdash^* ([E, \epsilon])$ y en consecuencia $([\alpha C, aw_1 bw_2] \vdash^* ([E, \epsilon])$.
- Si $\langle C, E \rangle[\alpha] \Rightarrow b \langle F', F \rangle[\alpha] a \langle G, E \rangle[\] \xrightarrow{m_1} bw_1 a \langle G, E \rangle[\] \xrightarrow{m_2} aw_1 bw_2$, existe un par de transiciones $C \xrightarrow{b} [C, F'$ y $C, [F \xrightarrow{a} G$, por hipótesis de inducción $(\alpha F', w_1] \vdash^* ([F, \epsilon])$ y $([G, w_2] \vdash^* ([E, \epsilon])$ y en consecuencia $([\alpha C, aw_1 bw_2] \vdash^* ([E, \epsilon])$.

□

Ejemplo 6.4 El autómata a pila embebido $(V_T, V_S, \Theta, \$_0, \$_f)$, donde $V_T = \{a, b, c, d\}$ y $V_S = \{\$, \$_f, B, C, D, E\}$, del ejemplo 6.2 acepta el lenguaje $\{a^n b^n c^n d^n \mid n \geq 0\}$. A partir de dicho autómata construiremos una gramática lineal de índices $(V_T, V_S \times V_S, V_S, \langle -, \$_0 \rangle, P)$. La tabla 6.13 muestra el conjunto de transiciones P , que ha sido obtenido a partir de las transiciones del autómata mostradas en la tabla 6.2. Para facilitar la lectura, hemos utilizado Γ para denotar cualquier posible elemento de V_S . La tabla 6.14 muestra la derivación de la cadena $aabbccdd$ en esta gramática. La primera columna muestra la producción aplicada para obtener la forma sentencial de la segunda columna. ¶

Corolario 6.7 *La clase de los lenguajes aceptados por los autómatas a pila embebidos coincide con la clase de los lenguajes de adjunción de árboles.*

Demostración:

Inmediata a partir de los teoremas 6.5 y 6.5. □

6.9. Tabulación

La ejecución directa de un autómata a pila embebido puede tener complejidad de orden exponencial con respecto al tamaño de la cadena de entrada. Ello se debe a que en el caso

- (a) $\langle \$_0, \$_0 \rangle[\circ\circ] \rightarrow \langle D, \$_0 \rangle[\circ\circ]$
(c) $\langle B, \Gamma \rangle[\circ\circ] \rightarrow \langle D, \Gamma \rangle[\circ\circ B]$
- (d) $\langle B, \Gamma \rangle[\circ\circ] \rightarrow \langle C, \Gamma \rangle[\circ\circ B]$
(f) $\langle E, \Gamma \rangle[\circ\circ B] \rightarrow \langle C, \Gamma \rangle[\circ\circ]$
- (g) $\langle C, \Gamma \rangle[\circ\circ] \rightarrow b \langle E, C \rangle[\circ\circ] \ c \langle C, \Gamma \rangle[]$
- (h) $\langle D, \Gamma \rangle[\circ\circ] \rightarrow a \langle B, C \rangle[\circ\circ] \ d \langle D, \Gamma \rangle[]$
(i) $\langle D, \Gamma \rangle[\circ\circ] \rightarrow a \langle B, D \rangle[\circ\circ] \ d \langle D, \Gamma \rangle[]$
(j) $\langle D, \Gamma \rangle[\circ\circ] \rightarrow \langle \$_f, \Gamma \rangle[\circ\circ]$
- (k) $\langle \$_0, \$_0 \rangle \rightarrow \epsilon$
(l) $\langle C, C \rangle \rightarrow \epsilon$
(m) $\langle D, D \rangle \rightarrow \epsilon$
(n) $\langle D, \$_f \rangle \rightarrow \epsilon$

Tabla 6.13: Producciones de la LIG derivada del EPDA que acepta $\{a^n b^n c^n d^n\}$

- (a) $\langle \$_0, \$_0 \rangle[] \Rightarrow \langle \$_0, \$_0 \rangle[] \ \langle D, \$_f \rangle[]$
(k) $\Rightarrow \langle D, \$_f \rangle[]$
(i) $\Rightarrow a \langle B, D \rangle[] \ d \langle D, \$_f \rangle[]$
(c) $\Rightarrow a \langle D, D \rangle[B] \ d \langle D, \$_f \rangle[]$
(h) $\Rightarrow aa \langle B, C \rangle[B] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(d) $\Rightarrow aa \langle C, C \rangle[BB] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(g) $\Rightarrow aab \langle E, C \rangle[BB] \ c \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(f) $\Rightarrow aab \langle C, C \rangle[B] \ c \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle d, \$_f \rangle[]$
(g) $\Rightarrow aabb \langle E, C \rangle[B] \ cc \langle C, C \rangle[] \ \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(f) $\Rightarrow aabb \langle C, C \rangle[] \ cc \langle C, C \rangle[] \ \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(l) $\Rightarrow aabbcc \langle C, C \rangle[] \ \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(l) $\Rightarrow aabbcc \langle C, C \rangle[] \ \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(l) $\Rightarrow aabbcc \langle D, D \rangle[] \ dd \langle D, \$_f \rangle[]$
(m) $\Rightarrow aabbccdd \langle D, \$_f \rangle[]$
(n) $\Rightarrow aabbccdd$

Tabla 6.14: Derivación de la cadena $aabbccdd$

Transición	Compilación de LIG	Compilación de TAG
$C \xrightarrow{a} F$	[SEL][PUB][SCAN]	[SEL][PUB][SCAN]
$C \mapsto C, [F$	[INIT][CALL]	[INIT][CALL]
$C \mapsto [C, F$	[SCALL-1]	[SCALL]
$C, [F \mapsto G$	[RET]	[RET][SRET][ARET][FRET]
$C \mapsto [C, X F$	[SCALL-2]	[ACALL]
$X C \mapsto [C, F$	[SCALL-3]	[FCALL]

Tabla 6.15: Tipos de transiciones de los EPDA

de que varias transiciones sean aplicables en una configuración dada, el contenido de la pila del autómata deberá replicarse y cada transición deberá aplicarse sobre una copia diferente. Para conseguir una complejidad polinomial deberemos evitar replicar la pila del autómata. Un modo de conseguirlo consiste en diseñar una técnica de tabulación que permita trabajar con representaciones condensadas de las configuraciones en lugar de con las configuraciones completas. Dichas representaciones condensadas se denominan ítems y se almacenan en una tabla, por lo que pueden ser reutilizadas y compartidas.

El diseño de una técnica de tabulación general para EPDA se presenta como una tarea complicada. Sin embargo, es posible diseñar una técnica de tabulación para el subconjunto de los EPDA que utilizan los siguientes tipos de transiciones:

- SWAP
- WRAP-A
- WRAP-B
- UNWRAP
- $C \mapsto [C, X F$
- $X C \mapsto [C, F$

Las transiciones $C \mapsto [C, X F$ pueden verse como la actuación combinada de una transición WRAP-B y una transición PUSH. Las transiciones $X C \mapsto [C, F$ pueden verse como la actuación combinada de una transición WRAP-B y una transición POP. En consecuencia, toda operación de apilamiento, que se realiza conjuntamente con una transición WRAP-B, deberá estar ligada con una transición WRAP-B en la que se realiza la extracción del elemento apilado. Es precisamente esta relación entre apilamientos y extracción de la pila la que nos permitirá diseñar una técnica de tabulación para EPDA.

El conjunto de transiciones seleccionado es suficiente para definir los esquemas de compilación para TAG y LIG de las secciones precedentes. La tabla 6.15 muestra los tipos de transiciones y su relación con las reglas de dichos esquemas de compilación. En esta tabla hemos considerado que las transiciones de tipo SWAP son las únicas que consumen terminales de la cadena de entrada. Este hecho no resta generalidad ya que, para cualquier transición, el efecto de la lectura de la cadena de entrada se puede conseguir mediante la aplicación consecutiva de una transición WRAP-A, una transición SWAP y una transición UNWRAP.

El primer paso en la definición de los ítems consiste en definir los distintos tipos de derivaciones que se pueden dar. En concreto, consideraremos los tres tipos siguientes:

Derivaciones de llamada. Son aquellas derivaciones en las que una pila es transmitida mediante transiciones de tipo WRAP-B, por lo que presentan la forma

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \end{aligned}$$

donde $A, B, C, X \in V_S$, $\alpha \in V_S^*$, $\Upsilon, \Upsilon_1 \in ([V_S^*])^*$ y no existe un par $([\alpha X F, f]) \neq ([\alpha X B, i])$ tal que

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X F, a_{f+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \end{aligned}$$

Las dos ocurrencias de α se refieren a la misma pila en el sentido de que dicha pila es transmitida sin cambios a través de la derivación. En este tipo de derivaciones se cumple que para cualquier $\Upsilon' \in ([V_S^*])^*$ y $\alpha' \in V_S^*$

$$\begin{aligned} (\Upsilon' [\alpha' A, a_{h+1} \dots a_n]) &\stackrel{*}{\vdash} (\Upsilon' [A \ \Upsilon_1 [\alpha' X B, a_{i+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon' [A \ \Upsilon_1 [\alpha' X C, a_{j+1} \dots a_n]) \end{aligned}$$

tal y como se indica en la figura 6.7. La independencia de la derivación con respecto a Υ y α permite que sean representadas por *ítems de llamada* de la forma

$$[A, h \mid B, i, X, C, j, X \mid -, -, -, -]$$

Derivaciones de retorno Son derivaciones resultado de aplicar transiciones UNWRAP, por lo que presentan la forma

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [B \ \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [B \ \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [C, a_{j+1} \dots a_n]) \end{aligned}$$

donde $A, B, C, D, E, X \in V_S$, $\alpha \in V_S^*$, $\Upsilon, \Upsilon_1, \Upsilon_2 \in ([V_S^*])^*$, α es transmitida sin cambios a lo largo de la derivación y no existen $([\alpha X F, f]) \neq ([\alpha X B, i])$ ni $([\alpha G, g]) \neq ([\alpha D, p])$ tal que

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X F, a_{f+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [B \ \Upsilon_2 [\alpha G, a_{g+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [B \ \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [B \ \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ &\stackrel{*}{\vdash} (\Upsilon [A \ \Upsilon_1 [C, a_{j+1} \dots a_n]) \end{aligned}$$

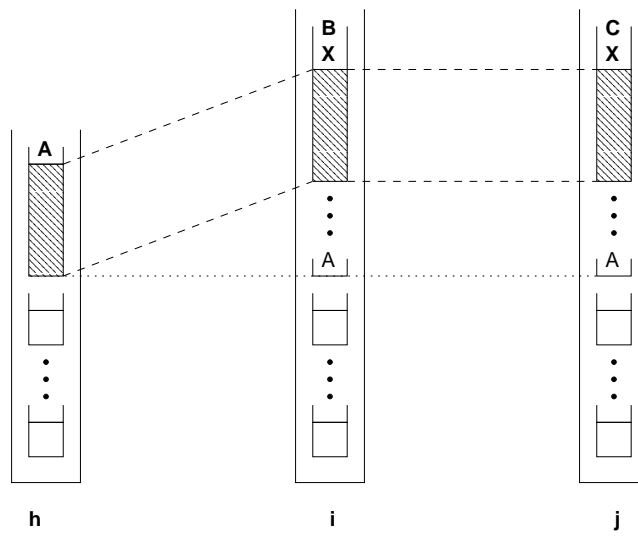


Figura 6.7: Derivaciones de llamada en EPDA

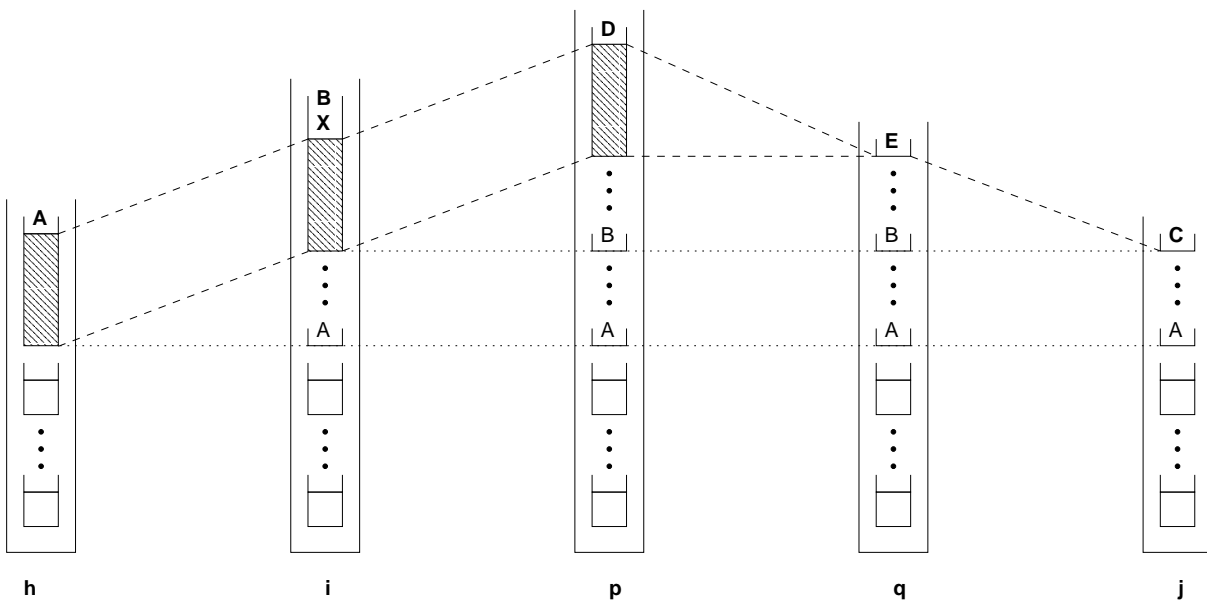


Figura 6.8: Derivaciones de retorno en EPDA

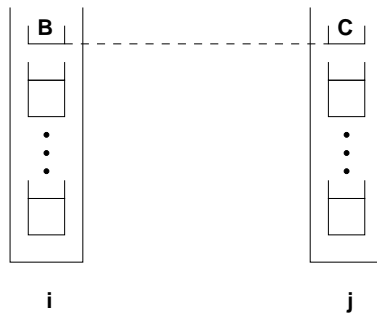


Figura 6.9: Derivaciones de puntos especiales en EPDA

En este tipo de derivaciones, para cualquier $\Upsilon' \in ([V_S^*])^*$ y $\alpha' \in V_S^*$ tal que existe una derivación $([\alpha'D, a_{p+1} \dots a_n])^* \vdash ([E, a_{q+1} \dots a_n])$, se cumple

$$\begin{aligned} (\Upsilon' [\alpha'A, a_{h+1} \dots a_n])^* &\vdash (\Upsilon [A \ \Upsilon_1 [\alpha'XB, a_{i+1} \dots a_n])^* \\ &\vdash (\Upsilon' [A \ \Upsilon_1 [B \ \Upsilon_2 [\alpha'D, a_{p+1} \dots a_n])^* \\ &\vdash (\Upsilon' [A \ \Upsilon_1 [B \ \Upsilon_2 [E, a_{q+1} \dots a_n])^* \\ &\vdash (\Upsilon' [A \ \Upsilon_1 [C, a_{j+1} \dots a_n])^* \end{aligned}$$

tal y como se indica en la figura 6.8. La independencia con respecto a Υ permite que este tipo de derivaciones sean representadas por *ítems de retorno* de la forma

$$[A, h \mid B, i, X, C, j, - \mid D, p, E, q]$$

en los que la parte (D, p, E, q) permite asegurar la relación entre $[\alpha D$ y $[\alpha A$.

Derivaciones de puntos especiales. Son aquellas derivaciones que sitúan una pila con un solo elemento en la cima de la pila principal, por lo que tienen la forma.

$$(\Upsilon [B, a_{i+1} \dots a_n])^* \vdash (\Upsilon [C, a_{j+1} \dots a_n])^*$$

donde $B, C \in V_S$ y $\Upsilon \in ([V_S^*])^*$ y no existe $([F, f]) \neq ([B, i])$ tal que

$$(\Upsilon [F, a_{f+1} \dots a_n])^* \vdash (\Upsilon [B, a_{i+1} \dots a_n])^* \vdash (\Upsilon [C, a_{j+1} \dots a_n])^*$$

Para cualquier $\Upsilon' \in ([V_S^*])^*$ se cumple que

$$(\Upsilon' [B, a_{i+1} \dots a_n])^* \vdash (\Upsilon' [C, a_{j+1} \dots a_n])^*$$

tal y como se indica en la figura 6.9, por lo que pueden ser representadas de modo condensado por *ítems de puntos especiales* de la forma

$$[-, - \mid B, i, -, C, j, - \mid -, -, -, -]$$

Los diferentes ítems se combinan entre sí a partir del ítem inicial

$$[-, - \mid \$_0, 0, -, \$_0, 0, - \mid -, -, -, -]$$

mediante las reglas de combinación de las tablas 6.16 y 6.17. La aceptación de la cadena de entrada $a_1 \dots a_n$ por parte del autómata se indica mediante la presencia de ítems finales de la forma

$$[-, - \mid F, 0, -, \$_f, n, - \mid -, -, -, -]$$

tal que existe una transición $\$_0 \mapsto [\$_0, F$ o una transición $\$_0 \mapsto \$_0, [F$.

Teorema 6.8 *La manipulación de configuraciones mediante la aplicación de transiciones en los autómatas a pila embebidos es equivalente a la manipulación de ítems mediante las reglas de combinación de las tablas 6.16 y 6.17.*

$$\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[A, h \mid B, i, X, F, k, X \mid -, -, -, -]} C \xrightarrow{a} F, \quad k = j \text{ si } a = \epsilon, \quad k = j + 1 \text{ si } a \in V_T$$

$$\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[A, h \mid B, i, X, F, k, - \mid D, p, E, q]} C \xrightarrow{a} F, \quad k = j \text{ si } a = \epsilon, \quad k = j + 1 \text{ si } a \in V_T$$

$$\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} C \mapsto C, [F]$$

$$\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} C^{[\infty]} \mapsto C^{[\infty]} F[]$$

$$\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[A, h \mid F, j, X, F, j, X \mid -, -, -, -]} C \mapsto [C, F]$$

$$\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} C \mapsto [C, F]$$

$$\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[C, j \mid F, j, X', F, j, X' \mid -, -, -, -]} C \mapsto [C, X'F]$$

$$\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[C, j \mid F, j, X', F, j, X' \mid -, -, -, -]} C \mapsto [C, X'F]$$

$$\frac{\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[M, m \mid N, t, X', A, h, X' \mid -, -, -, -]}}{[M, m \mid F, j, X', F, j, X' \mid -, -, -, -]} XC \mapsto [C, F]$$

$$\frac{\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[M, m \mid N, t, X', A, h, - \mid D, p, E, q]}}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} XC \mapsto [C, F]$$

Tabla 6.16: Combinación de ítems en EPDA (fase de llamada)

$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, X \mid -, -, -, -]} \quad \begin{array}{l} C \mapsto C, [F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, - \mid D, p, E, q] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto C, [F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [A, h \mid F', j, X, F, k, - \mid D, p, E, q] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto [C, F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, - \mid D, p, E, q] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto [C, F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [C, j \mid F', j, X', F, k, - \mid D, p, E, q] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\ [A, h \mid D, p, X, E, q, - \mid O, u, P, v] \end{array}}{[A, h \mid B, i, X, G, k, - \mid O, u, P, v]} \quad \begin{array}{l} C \mapsto [C, X'F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [C, j \mid F', j, X', F, k, - \mid O, u, P, v] \\ [A, h \mid B, i, X, C, j, - \mid D, p, E, q] \\ [-, - \mid O, u, -, P, v, - \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto [C, X'F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [M, m \mid F', j, X', F, k, - \mid D, p, E, q] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\ [M, m \mid N, t, X', A, h, X' \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, - \mid F', j, F, k]} \quad \begin{array}{l} XC \mapsto [C, F'] \\ C, [F \mapsto G] \end{array}$
$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\ [M, m \mid N, t, X', A, h, - \mid D, p, E, q] \end{array}}{[A, h \mid B, i, X, G, k, - \mid F', j, F, k]} \quad \begin{array}{l} XC \mapsto [C, F'] \\ C, [F \mapsto G] \end{array}$

Tabla 6.17: Combinación de ítems en EPDA (fase de retorno)

Demostración:

Debemos demostrar que para toda derivación existe un ítem que la representa y que para todo ítem existe una derivación a la que representa, teniendo en cuenta las diferentes formas en las que las derivaciones y los ítems pueden haber sido producidos. Con el fin de abreviar, realizaremos ambas demostraciones en paralelo. Para ello será necesario hacer una lista exhaustiva con todos los tipos de derivaciones que se pueden dar, mostrando para cada una de estas derivaciones la regla de combinación de ítems correspondiente. Dicha lista es la que se muestra a continuación.

- Derivaciones que son el resultado de aplicar una transición $C \xrightarrow{a} F$

- a una derivación de llamada:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X F, a_{k+1} \dots a_n]) \end{array} \\ \frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[A, h \mid B, i, X, F, k, X \mid -, -, -, -]} C \xrightarrow{a} F, \quad k = j \text{ si } a = \epsilon, \quad k = j + 1 \text{ si } a \in V_T \end{array}$$

- a una derivación de retorno:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [F, a_{k+1} \dots a_n]]) \end{array} \\ \frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[A, h \mid B, i, X, F, k, - \mid D, p, E, q]} C \xrightarrow{a} F, \quad k = j \text{ si } a = \epsilon, \quad k = j + 1 \text{ si } a \in V_T \end{array}$$

- a una derivación de puntos especiales:

$$\begin{array}{l} (\Upsilon [B, a_{i+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [F, a_{k+1} \dots a_n]) \end{array} \\ \frac{[-, - \mid B, i, -, C, j, - \mid -, -, -, -]}{[-, - \mid B, i, -, F, k, - \mid -, -, -, -]} C \xrightarrow{a} F, \quad k = j \text{ si } a = \epsilon, \quad k = j + 1 \text{ si } a \in V_T \end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C \mapsto C, [F$

- a una derivación de llamada:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X C F, a_{j+1} \dots a_n]]) \end{array} \\ \frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} C \mapsto C, [F \end{array}$$

- a una derivación de retorno:

$$\begin{array}{l} (\Upsilon [A[\delta], a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [F, a_{j+1} \dots a_n]]) \end{array} \\ \frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} C \mapsto C, [F \end{array}$$

- a una derivación de puntos especiales:

$$\begin{array}{c}
(\Upsilon [B, a_{i+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [C [F, a_{j+1} \dots a_n]) \end{array} \\
\frac{[-, - \mid B, i, -, C, j, - \mid -, -, -, -]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} \quad C \mapsto C, [F]
\end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C \mapsto [C, F$

- a una derivación de llamada:

$$\begin{array}{c}
(\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \\ ; \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \end{array} \\
\vdash (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \\
\vdash (\Upsilon [A \Upsilon_1 [C [\Upsilon_1 [\alpha X F, a_{j+1} \dots a_n]) \\
\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[A, h \mid F, j, X, F, j, X \mid -, -, -, -]} \quad C \mapsto [C, F]
\end{array}$$

- a una derivación de retorno:

$$\begin{array}{c}
(\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [F, a_{j+1} \dots a_n]) \end{array} \\
\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} \quad C \mapsto [C, F]
\end{array}$$

- a una derivación de puntos especiales:

$$\begin{array}{c}
(\Upsilon [B, a_{i+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [C [F, a_{j+1} \dots a_n]) \end{array} \\
\frac{[-, - \mid B, i, -, C, j, - \mid -, -, -, -]}{[-, - \mid F, j, -, F, j, - \mid -, -, -, -]} \quad C \mapsto [C, F]
\end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C \mapsto [C, X'F$

- a una derivación de llamada:

$$\begin{array}{c}
(\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [\alpha X X' F, a_{j+1} \dots a_n]) \end{array} \\
\frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[C, j \mid F, j, X', F, j, X' \mid -, -, -, -]} \quad C \mapsto [C, X'F]
\end{array}$$

- a una derivación de retorno:

$$\begin{array}{c}
(\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [X' F, a_{j+1} \dots a_n]) \end{array} \\
\frac{[A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[C, j \mid F, j, X', F, j, X' \mid -, -, -, -]} \quad C \mapsto [C, X'F]
\end{array}$$

- a una derivación de puntos especiales:

$$\begin{array}{c} (\Upsilon [B, a_{i+1} \dots a_n]) \quad * \\ \vdash (\Upsilon [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [C [X'F, a_{j+1} \dots a_n]]) \\ \hline \frac{[-, - \mid B, i, -, C, j, - \mid -, -, -, -]}{[C, j \mid F, j, X', F, j, X' \mid -, -, -, -]} \quad C \mapsto [C, X'F \end{array}$$

- Derivaciones que son el resultado de aplicar una transición $XC \mapsto [C, F$ a una derivación de llamada, con los tres casos siguientes:

- la derivación de llamada es a su vez obtenida a partir de una derivación de llamada:

$$\begin{array}{c} (\Upsilon [\alpha M, a_{m+1} \dots a_n]) \quad * \\ \vdash (\Upsilon [M \Upsilon_1 [\alpha X'N, a_{t+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [\alpha X'A, a_{h+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_1 [\alpha X'XB, a_{i+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_1 [\alpha X'XC, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_1 [C [\alpha X'F, a_{j+1} \dots a_n]]) \\ \hline \frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[M, m \mid N, t, X', A, h, X' \mid -, -, -, -]} \quad XC \mapsto [C, F \\ \hline [M, m \mid F, j, X', F, j, X' \mid -, -, -, -] \end{array}$$

- la derivación de llamada es a su vez obtenida a partir de una derivación de retorno:

$$\begin{array}{c} (\Upsilon [\alpha M, a_{m+1} \dots a_n]) \quad * \\ \vdash (\Upsilon [M \Upsilon_1 [\alpha X'N, a_{t+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [N \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [N \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A, a_{h+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A [XB, a_{i+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A [XC, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [M \Upsilon_1 [A [C [F, a_{j+1} \dots a_n]]) \\ \hline \frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[M, m \mid N, t, X', A, h, - \mid D, p, E, q]} \quad XC \mapsto [C, F \\ \hline [-, - \mid F, j, -, F, j, - \mid -, -, -, -] \end{array}$$

- la derivación de llamada es a su vez obtenida a partir de una derivación de puntos especiales:

$$\begin{array}{c} (\Upsilon [N, a_{t+1} \dots a_n]) \quad * \\ \vdash (\Upsilon [A, a_{h+1} \dots a_n]) \\ \vdash (\Upsilon [A [XB, a_{i+1} \dots a_n]]) \\ \vdash (\Upsilon [A [XC, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A [C [F, a_{j+1} \dots a_n]]) \\ \hline \frac{[A, h \mid B, i, X, C, j, X \mid -, -, -, -]}{[-, - \mid N, t, -, A, h, - \mid -, -, -, -]} \quad XC \mapsto [C, F \\ \hline [-, - \mid F, j, -, F, j, - \mid -, -, -, -] \end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C, [F \mapsto G$ a una derivación obtenida tras aplicar una transición $C \mapsto C, [F'$

- a una derivación de llamada:

$$\begin{array}{c} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha XB, a_{i+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha XC, a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha XC [F', a_{j+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha XC [F, a_{k+1} \dots a_n]]) \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha XG, a_{k+1} \dots a_n]]) \end{array}$$

$$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, X \mid -, -, -, -]} \quad \begin{array}{l} C \mapsto C, [F'] \\ C, [F \mapsto G] \end{array}$$

- a una derivación de retorno:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [F', a_{j+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C [F, a_{k+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [G, a_{k+1} \dots a_n]) \end{array} \end{array}$$

$$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [A, h \mid B, i, X, C, j, - \mid D, p, E, q] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto C, [F'] \\ C, [F \mapsto G] \end{array}$$

- a una derivación de puntos especiales:

$$\begin{array}{l} (\Upsilon [B, a_{i+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [C [F', a_{j+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [C [F, a_{k+1} \dots a_n]) \\ \vdash (\Upsilon [G, a_{k+1} \dots a_n]) \end{array} \end{array}$$

$$\frac{\begin{array}{l} [-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\ [-, - \mid B, i, X, C, j, - \mid -, -, -, -] \end{array}}{[-, - \mid B, i, X, G, k, - \mid -, -, -, -]} \quad \begin{array}{l} C \mapsto C, [F'] \\ C, [F \mapsto G] \end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C, [F \mapsto G]$ a una derivación obtenida tras aplicar una transición $C \mapsto [C, F']$

- a una derivación de llamada:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [\alpha X F', a_{j+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C [F, a_{k+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [G, a_{k+1} \dots a_n]) \end{array} \end{array}$$

$$\frac{\begin{array}{l} [A, h \mid F', j, X, F, k, - \mid D, p, E, q] \\ [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \end{array}}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad \begin{array}{l} C \mapsto [C, F'] \\ C, [F \mapsto G] \end{array}$$

- a una derivación de retorno:

$$\begin{array}{l} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) \quad \begin{array}{l} * \\ \vdash (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C \Upsilon_2 [E, a_{q+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [C [F', a_{j+1} \dots a_n]) \\ * \\ \vdash (\Upsilon [A \Upsilon_1 [C [F, a_{k+1} \dots a_n]) \\ \vdash (\Upsilon [A \Upsilon_1 [G, a_{k+1} \dots a_n]) \end{array} \end{array}$$

$$\frac{[-, - \mid F', j, -, F, k, - \mid -, -, -, -] \quad [A, h \mid B, i, X, C, j, - \mid D, p, E, q]}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad C \mapsto [C, F'] \quad C, [F \mapsto G]$$

- a una derivación de puntos especiales:

$$\begin{aligned} (\Upsilon [B, a_{i+1} \dots a_n]) & \overset{*}{\vdash} (\Upsilon [C, a_{j+1} \dots a_n]) \\ & \vdash (\Upsilon [C [F', a_{j+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [C [F, a_{k+1} \dots a_n]]) \\ & \vdash (\Upsilon [G, a_{k+1} \dots a_n]) \end{aligned}$$

$$\frac{[-, - \mid F', j, -, F, k, - \mid -, -, -, -] \quad [-, - \mid B, i, -, C, j, - \mid -, -, -, -]}{[-, - \mid B, i, -, G, k, - \mid -, -, -, -]} \quad C \mapsto [C, F'] \quad C, [F \mapsto G]$$

- Derivaciones que son el resultado de aplicar una transición $C, [F \mapsto G]$ a una derivación obtenida tras aplicar una transición $C \mapsto [C, X'F']$

- a una derivación de llamada:

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [\alpha X C, a_{j+1} \dots a_n]]) \\ & \vdash (\Upsilon [A \Upsilon_1 [C [\alpha X X' F', a_{j+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [\alpha X D, a_{p+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [D \Upsilon_3 [\alpha O, a_{u+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [D \Upsilon_3 [P, a_{v+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F, a_{k+1} \dots a_n]]) \\ & \vdash (\Upsilon [A \Upsilon_1 [G, a_{k+1} \dots a_n]) \end{aligned}$$

$$\frac{[C, j \mid F', j, X', F, k, - \mid D, p, E, q] \quad [A, h \mid B, i, X, C, j, X \mid -, -, -, -] \quad [A, h \mid D, p, X, E, q, - \mid O, u, P, v]}{[A, h \mid B, i, X, G, k, - \mid O, u, P, v]} \quad C \mapsto [C, X'F'] \quad C, [F \mapsto G]$$

- a una derivación de retorno:

$$\begin{aligned} (\Upsilon [\alpha A, a_{h+1} \dots a_n]) & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [\alpha X B, a_{i+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [B \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C, a_{j+1} \dots a_n]]) \\ & \vdash (\Upsilon [A \Upsilon_1 [C [X'F', a_{j+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_3 [O, a_{u+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F' \Upsilon_3 [P, a_{v+1} \dots a_n]]) \\ & \overset{*}{\vdash} (\Upsilon [A \Upsilon_1 [C [F, a_{k+1} \dots a_n]]) \\ & \vdash (\Upsilon [A \Upsilon_1 [G, a_{k+1} \dots a_n]) \end{aligned}$$

$$\frac{[C, j \mid F', j, X', F, k, - \mid O, u, P, v] \quad [A, h \mid B, i, X, C, j, - \mid D, p, E, q] \quad [-, - \mid O, u, -, P, v, - \mid -, -, -, -]}{[A, h \mid B, i, X, G, k, - \mid D, p, E, q]} \quad C \mapsto [C, X'F'] \quad C, [F \mapsto G]$$

- a una derivación de puntos especiales:

$$\begin{array}{l}
(\Upsilon [B, a_{i+1} \dots a_n]) \quad \begin{array}{l}
^* (\Upsilon [C, a_{j+1} \dots a_n]) \\
\vdash (\Upsilon [C [X'F', a_{j+1} \dots a_n]) \\
^* (\Upsilon [C [F' \Upsilon_1 [O, a_{u+1} \dots a_n]]) \\
^* (\Upsilon [C [F' \Upsilon_1 [P, a_{v+1} \dots a_n]]) \\
^* (\Upsilon [C [F, a_{k+1} \dots a_n]]) \\
\vdash (\Upsilon [G, a_{k+1} \dots a_n])
\end{array}
\end{array}$$

$$\frac{
\begin{array}{l}
[C, j \mid F', j, X', F, k, - \mid O, u, P, v] \\
[-, - \mid B, i, -, C, j, - \mid -, -, -, -] \\
[-, - \mid O, u, -, P, v, - \mid -, -, -, -]
\end{array}
}{
\begin{array}{l}
[-, - \mid B, i, -, G, k, - \mid -, -, -, -]
\end{array}
}
\quad
\begin{array}{l}
C \mapsto [C, X'F'] \\
C, [F \mapsto G]
\end{array}$$

- Derivaciones que son el resultado de aplicar una transición $C, [F \mapsto G$ a una derivación obtenida tras aplicar una transición $XC \mapsto [C, F'$ a una derivación de llamada, con los tres casos siguientes:

- la derivación de llamada es a su vez derivada de una derivación de llamada:

$$\begin{array}{l}
(\Upsilon [\alpha M, a_{m+1} \dots a_n]) \quad \begin{array}{l}
^* (\Upsilon [M \Upsilon_1 \alpha X'N, a_{t+1} \dots a_n]) \\
^* (\Upsilon [M \Upsilon_1 [\alpha X'A, a_{h+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [\alpha X'XB, a_{i+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [\alpha X'XC, a_{j+1} \dots a_n]]) \\
\vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [C [\alpha X'F', a_{j+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [C [F' \Upsilon_3 [D[\delta], a_{p+1} \dots a_n]]) \\
\vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [C [F' \Upsilon_3 [E, a_{q+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [C [F, a_{k+1} \dots a_n]]) \\
\vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_2 [G, a_{k+1} \dots a_n]])
\end{array}
\end{array}$$

$$\frac{
\begin{array}{l}
[M, m \mid F', j, X', F, k, - \mid D, p, E, q] \\
[A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\
[M, m \mid N, t, X', A, h, X' \mid -, -, -, -]
\end{array}
}{
\begin{array}{l}
[A, h \mid B, i, X, G, k, - \mid F', j, F, k]
\end{array}
}
\quad
\begin{array}{l}
XC \mapsto [C, F'] \\
C, [F \mapsto G]
\end{array}$$

- la derivación de llamada es a su vez derivada de una derivación de retorno:

$$\begin{array}{l}
(\Upsilon [\alpha M, a_{m+1} \dots a_n]) \quad \begin{array}{l}
^* (\Upsilon [M \Upsilon_1 [\alpha X'N, a_{t+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [N \Upsilon_2 [\alpha D, a_{p+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [N \Upsilon_2 [E, a_{q+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A, a_{h+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_3 [XB, a_{i+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_3 [XC, a_{j+1} \dots a_n]]) \\
\vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_3 [XC [F', a_{j+1} \dots a_n]]) \\
^* (\Upsilon [M \Upsilon_1 [A \Upsilon_3 [XC [F, a_{k+1} \dots a_n]]) \\
\vdash (\Upsilon [M \Upsilon_1 [A \Upsilon_3 [G, a_{k+1} \dots a_n]])
\end{array}
\end{array}$$

$$\frac{
\begin{array}{l}
[-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\
[A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\
[M, m \mid N, t, X', A, h, - \mid D, p, E, q]
\end{array}
}{
\begin{array}{l}
[A, h \mid B, i, X, G, k, - \mid F', j, F, k]
\end{array}
}
\quad
\begin{array}{l}
XC \mapsto [C, F'] \\
C, [F \mapsto G]
\end{array}$$

- la derivación de llamada es a su vez derivada de una derivación de puntos especiales:

$$\begin{array}{c}
([\alpha X'N, a_{t+1} \dots a_n) \quad \begin{array}{l} * \\ \vdash ([A, a_{h+1} \dots a_n) \\ * \\ \vdash ([A \Upsilon_3 [XB, a_{i+1} \dots a_n) \\ * \\ \vdash ([A \Upsilon_3 [XC, a_{j+1} \dots a_n) \\ \vdash ([A \Upsilon_3 [C [F', a_{j+1} \dots a_n) \\ * \\ \vdash ([A \Upsilon_3 [C [F, a_{k+1} \dots a_n) \\ \vdash ([A \Upsilon_3 [G, a_{k+1} \dots a_n) \end{array} \\
\\
\frac{
\begin{array}{l}
[-, - \mid F', j, -, F, k, - \mid -, -, -, -] \\
[A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\
[-, - \mid N, t, X', A, h, - \mid -, -, -, -]
\end{array}
}{
[A, h \mid B, i, X, G, k, - \mid F', j, F, k]
}
\quad
\begin{array}{l}
XC \mapsto [C, F' \\
C, [F \mapsto G
\end{array}
\end{array}$$

□

La complejidad espacial de la técnica de tabulación con respecto a la longitud n de la cadena de entrada es $\mathcal{O}(n^5)$, puesto que cada ítem almacena 5 posiciones de la cadena de entrada. La complejidad temporal es $\mathcal{O}(n^7)$. Dicha complejidad viene dada por la siguiente regla:

$$\frac{
\begin{array}{l}
[C, j \mid F', j, X', F, k, - \mid D, p, E, q] \\
[A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\
[A, h \mid D, p, X, E, q, - \mid O, u, P, v]
\end{array}
}{
[A, h \mid B, i, X, G, k, - \mid O, u, P, v]
}
\quad
\begin{array}{l}
C \mapsto [C, X'F' \\
C, [F \mapsto G
\end{array}$$

Podemos reducir la complejidad temporal aplicando la técnica propuesta en [53, 125], según la cual la regla anterior debe ser dividida en dos reglas de menor complejidad, de tal modo que el consecuente de la primera sea un pseudo-ítem intermedio que proporcione a la segunda regla la información necesaria para que la aplicación combinada de ambas reglas sea equivalente a la regla original. En el caso que nos ocupa, las dos reglas obtenidas son

$$\frac{
\begin{array}{l}
[C, j \mid F', j, X', F, k, - \mid D, p, E, q] \\
[A, h \mid D, p, X, E, q, - \mid O, u, P, v]
\end{array}
}{
[[F', j, X', F, k, - \mid O, u, P, v]]
}
\quad
\begin{array}{l}
C \mapsto [C, X'F' \\
C, [F \mapsto G
\end{array}$$

$$\frac{
\begin{array}{l}
[[F', j, X', F, k, - \mid O, u, P, v]] \\
[A, h \mid B, i, X, C, j, X \mid -, -, -, -] \\
[A, h \mid D, p, X, E, q, - \mid O, u, P, v]
\end{array}
}{
[A, h \mid B, i, X, G, k, - \mid O, u, P, v]
}
\quad
\begin{array}{l}
C \mapsto [C, X'F' \\
C, [F \mapsto G
\end{array}$$

donde $[[F', j, X', F, k, - \mid O, u, P, v]]$ es el pseudo-ítem intermedio que relaciona las dos reglas. La primera regla ignora la posición h , que es posteriormente recuperada del segundo y tercer ítem que intervienen en la segunda regla, que junto con el pseudo-ítem son suficientes para garantizar la existencia del ítem $[C, j \mid F', j, X', F, k, - \mid D, p, E, q]$ por la definición de derivaciones de llamada y retorno. La primera regla presenta una complejidad temporal $\mathcal{O}(n^6)$ (la posición h no interviene) y la segunda presenta también una complejidad $\mathcal{O}(n^6)$ (las posiciones p y q no intervienen) por lo que hemos logrado rebajar la complejidad temporal final de la técnica de tabulación a $\mathcal{O}(n^6)$.